

UNIVERSIDAD DE CUENCA



FACULTAD DE INGENIERÍA

ESCUELA DE INFORMÁTICA

“Visualización y Explotación de información geoespacial en formato RDF (Resource Description Framework)”

Tesis previa a la obtención del título de Ingeniero de Sistemas

Autor: Jonnathan Ramiro Ambrosi Moreira

C.I: 0104613591

Directora: Ing. Rosa Lucía Lupercio Novillo, PhD

C.I: 0102698693

Co-Director: Ing. Víctor Hugo Saquicela Galarza, PhD

C.I: 0103599577

Cuenca – Ecuador

2017



Resumen

El desarrollo de la Web Semántica significa un cambio de paradigma, puesto que propone el cambio de una web sintáctica a una web cuyo principal objetivo es proveer de significado a los datos. Una aplicación de la Web Semántica es Linked Data que se puede definir como un conjunto de buenas prácticas para publicar y enlazar datos. Con Linked Data tanto personas y maquinas pueden explorar la web de los datos enlazados.

En este sentido, existe la iniciativa Ecuatoriana de datos enlazados geoespaciales denominada GEO LINKED DATA ECUADOR. Actualmente esta iniciativa hace uso de la herramienta MAP4RDF para la visualización y explotación de información geoespacial, con lenguaje de consulta SPARQL y la capacidad de consultas espaciales con GeoSPARQL. Mediante el uso del estándar GeoSPARQL es posible representar y consultar información geoespacial de diferentes rasgos terrestres como edificios, vías, ríos, etc. MAP4RDF es una aplicación Web Java que puede ser cargada y desplegada sobre un Servidor Web, además permite visualizar fuentes de datos en formato RDF, esta aplicación presenta limitaciones como el excesivo tiempo que le toma en desplegar los recursos de forma gráfica sobre un mapa, además la imposibilidad de gestionar parámetros gráficos (color, ancho, transparencia e imagen de marcador).

En este trabajo se desarrolló un visor de información geoespacial para visualizar y explotar el repositorio de la iniciativa ecuatoriana de datos enlazados geoespaciales. Los usuarios a quienes va dirigida esta aplicación serán capaces de interactuar con ella sin necesidad de tener conocimientos de programación o Linked Data. El visor utiliza como lenguaje de consulta SPARQL y la capacidad de consultas geoespaciales de GeoSPARQL. El visor fue desplegado sobre el servidor linkeddata.ec del Departamento de Ciencias de la Computación (DCC) de la Universidad de Cuenca. Para el desarrollo de este trabajo se usó servicios tanto de la plataforma geoespacial CARTO (Sistema de información geográfico Web) como de la plataforma Apache Marmotta (Servidor de tripletas RDF). La aplicación Web desarrollada es capaz de visualizar recursos que residen sobre la base de datos geoespacial PostgreSQL de la plataforma CARTO (Visualización de recursos íntegros) así como recursos que residen sobre la base de datos PostgreSQL de la plataforma Apache Marmotta (Visualización de recursos producto de una consulta SPARQL).

Palabras clave: Web Semántica, Linked Data, GEO LINKED DATA ECUADOR, MAP4RDF, SPARQL, GeoSPARQL, RDF, CARTO, Servidor Web, Sistema de información geográfico Web, Servidor de tripletas RDF, Apache Marmotta, PostgreSQL, geoespacial, linkeddata.ec.



Abstract

The development of the Semantic Web means a paradigm shift, since it proposes the change from a syntactic web to a web whose main objective is to provide meaning to the data. A Semantic Web application is Linked Data that can be defined as a set of good practices for publishing and linking data. With Linked Data both people and machines can explore the web of linked data.

In this sense, there is the Ecuadorian initiative of linked geospatial data called GEO LINKED DATA ECUADOR. Currently this initiative makes use of the MAP4RDF tool for the visualization and exploitation of geospatial information, with SPARQL query language and the capacity of spatial queries with GeoSPARQL. By using the GeoSPARQL standard it is possible to represent and consult geospatial information of different terrestrial features such as buildings, roads, rivers, etc. MAP4RDF is a Java Web application that can be loaded and deployed on a Web Server, in addition to displaying data sources in RDF format, this application has limitations such as the excessive time it takes to deploy resources graphically on a map, in addition the inability to manage graphic parameters (color, width, transparency and marker image).

In this work, a geospatial information viewer was developed to visualize and exploit the Ecuadorian repository of geospatial linked data. Users targeted by this application will be able to interact with it without the need to have programming knowledge or Linked Data. The viewer uses the SPARQL query language and the geospatial query capability of GeoSPARQL. The viewer was deployed on the linkeddata.ec server of the Department of Computer Science (DCC) of the University of Cuenca. For the development of this work, services were used both from the geospatial platform CARTO (Web Geographic Information System) and from the Apache Marmotta platform (RDF Triple Store). The developed Web application is able to visualize resources that reside on the PostgreSQL geospatial database of the CARTO platform, as well as resources that reside on the PostgreSQL database of the Apache Marmotta platform.

Keywords: Semantic Web, Linked Data, GEO LINKED DATA ECUADOR, MAP4RDF, SPARQL, GeoSPARQL, RDF, CARTO, Web Server, Web Geographic Information System, RDF Triple Store, Apache Marmotta, PostgreSQL, geospatial, linkeddata.ec.



Índice general

Resumen	2
Abstract.....	3
Índice general.....	4
Índice de figuras	8
Índice de tablas.....	10
Índice de fragmentos de código.....	12
1. Introducción.....	17
1.1. Antecedentes.....	17
1.2. Planteamiento del problema	18
1.3. Justificación	19
1.4. Objetivos	20
1.4.1 Objetivo general.....	20
1.4.2 Objetivos específicos	20
1.5. Metodología	21
1.5.1. Búsqueda Bibliográfica.....	21
1.5.2. Clasificación Bibliográfica	21
1.5.3. Investigación Bibliográfica	21
1.5.4. Redacción.....	21
1.5.5. Elección de herramientas	21
1.5.6. Diseño.....	22
1.5.7. Implementación	22
1.5.8. Pruebas.....	22
1.5.9. Implantación.....	22
2. Conceptos previos	23
2.1. Web semántica	23
2.1.1. Definición de web semántica.....	23
2.1.2. RDF (Resource Description framework)	24



2.1.3. Ontología.....	24
2.1.4. SPARQL y GeoSPARQL.....	25
2.2. Datos enlazados y datos enlazados geoespaciales	28
2.2.1. Datos enlazados.....	28
2.2.2. Datos enlazados geoespaciales	29
2.3. Metodología para la generación y publicación de datos enlazados geoespaciales ...	29
2.3.1. Especificación	30
2.3.2. Modelado.....	30
2.3.3. Generación de RDF	30
2.3.4. Generación de Links	31
2.3.5. Publicación	32
2.3.6. Explotación	32
2.4. Plataformas de soporte de almacenamiento de RDF con información geoespacial	33
2.4.1. OpenLink Virtuoso	33
2.4.2. Apache Marmotta	34
2.4.3. Parliament	34
2.4.4. Fuseki	34
2.5. Plataformas de explotación de información geoespacial que soportan RDF y GeoSPARQL.....	35
2.5.1. Map4RDF	35
2.5.2. Sextant	35
2.5.3. SemMap	36
2.5.4. LINKEDGEODATABROWSER.....	36
2.6. Visualizadores genéricos de información geoespacial	37
2.6.1. Google Fusion Tables	37
2.6.2. GeoCommons.....	37
2.6.3. Carto.....	38
2.7. Herramientas de desarrollo y soporte	38
2.7.1. Apache Jena	39
2.7.2. GeoTools	39
2.7.3. CartoDB-Java-Client.....	40



2.7.4. Carto.js.....	40
2.7.5. Carto.css.....	41
3. Análisis y diseño de la solución	42
3.1. Análisis.....	42
3.1.1. Especificaciones	42
3.1.2. Requerimientos funcionales.....	42
3.1.3. Requerimientos no funcionales	45
3.1.4. Definición de los límites del sistema	46
3.1.5. Lenguaje de programación	46
3.2. Diseño.....	47
3.2.1. Diagrama casos de uso	47
3.2.2. Casos de uso en formato extendido.....	47
3.2.3. Diagrama de clases.....	52
3.2.4. Arquitectura del sistema	52
3.2.5. Diseño de interfaz	54
4. Implementación de la solución.....	55
4.1. Descripción de componentes de la solución.....	55
4.1.1. Descripción Base de datos geoespacial CARTO.....	55
4.1.2. Descripción Servidor de tripletas RDF Apache Marmotta.....	56
4.1.3. Descripción Script para importar RDF hacia CARTO	57
4.1.4. Descripción Visor de información geoespacial.	57
4.2. Script para importar RDF hacia CARTO	58
4.2.1. Entorno y lenguaje de desarrollo.....	58
4.2.2. Librerías necesarias.....	58
4.2.3. Funcionalidades.....	58
4.2.4. Parámetros de configuración.....	59
4.2.5. Prueba de parámetros de acceso a la plataforma CARTO	60
4.2.6. Consultas de selección	60
4.2.7. Importar RDF con información geoespacial hacia CARTO.....	61
4.2.8. Listar Recursos alojados sobre CARTO.....	61
4.2.9. Eliminar Recursos alojados sobre CARTO	62



4.3. Visor de información geoespacial.....	63
4.3.1. Entorno y lenguaje de desarrollo.....	63
4.3.2. Librerías necesarias.....	63
4.3.3. Funcionalidades.....	64
4.3.4. Parámetros de configuración.....	64
4.3.5. Visualizar información geoespacial desde CARTO	65
4.3.6. Visualizar información geoespacial producto de una consulta hacia un Sparql Endpoint.....	67
4.3.7. Gestión de parámetros gráficos	68
4.3.8. Borrar recursos desplegados sobre el mapa.....	69
5. Despliegue sobre linkeddata.ec	70
5.1. Configuración y Despliegue de Apache Marmotta.....	70
5.1.1. Obtención y construcción de la herramienta.....	70
5.1.2. Despliegue de la herramienta sobre linkeddata.ec.....	71
5.1.3. Configuración de la herramienta.....	72
5.1.4. Importación de archivos RDF hacia la herramienta.....	73
5.2. Configuración y Despliegue del Visor de información geoespacial	74
5.2.1 Configuración y construcción de la herramienta.....	74
5.2.2 Despliegue del visor sobre linkeddata.ec	75
5.3. Resultados.....	77
5.3.1. Visualización de recursos íntegros.....	77
5.3.2. Visualización de recursos producto de una consulta predeterminada	79
5.3.3. Gestionar parámetros gráficos.....	81
6. Conclusiones	83
Anexos	86
A. Importación de archivo <i>cantonesAzuaay.rdf</i> a CARTO.....	86
B. Eliminación de archivo <i>cantonesAzuaay.rdf</i> de CARTO.....	88
C. Enumeración de conjuntos de datos sobre CARTO.	89
Acrónimos.....	90
Bibliografía.....	92



Índice de figuras

Figura 1: Evolución de la Web (fuente: La web Social, http://e-educativa.catedu.es).....	23
Figura 2: Representación recurso Quito	24
Figura 3: Sintaxis Consulta SPARQL	26
Figura 4: Sintaxis Consulta SPARQL extensión GeoSPARQL.....	27
Figura 5: Extracto Nube de Datos Enlazados (fuente: Linked Open Data Diagram, http://lod-cloud.net/).....	28
Figura 6: Fases Metodología generación de Datos enlazados.....	30
Figura 7: Despliegue de información geoespacial sobre un mapa	33
Figura 8: Diagrama de casos de uso.....	47
Figura 9: Diagrama de clases.....	52
Figura 10: Arquitectura Visor de información geoespacial.....	53
Figura 11: Interfaz Visor de información geoespacial.....	54
Figura 12: Estructura base de datos plataforma CARTO.....	55
Figura 13: Extracto grafos almacenados sobre Apache Marmotta	56
Figura 14: Interfaz y operaciones de Script.....	57
Figura 15: Visor de información geoespacial contenida en RDF	57
Figura 16: Arquitectura Script para importar RDF hacia CARTO.....	58
Figura 17: Construcción de Apache Marmotta.....	71
Figura 18: Acceso al servidor de Aplicaciones Web	71
Figura 19: Despliegue de Apache Marmotta.....	71
Figura 20: Interfaz Apache Marmotta	72
Figura 21: Configuración de PostgreSQL Apache Marmotta.....	72
Figura 22: Configuración de Usuarios Apache Marmotta	73
Figura 23: Creación de grafos Apache Marmotta.....	73
Figura 24: Importar RDF Apache Marmotta	74
Figura 25: Construcción de Visor de Información geoespacial.....	75
Figura 26: Acceso al servidor de Aplicaciones Web	76
Figura 27: Despliegue de Visor de información geoespacial	76
Figura 28: Apertura del Visor de información geoespacial.....	76
Figura 29: Interfaz inicial Script de importación.....	86
Figura 30: Selección de archivo RDF	86
Figura 31: Lectura y validación de RDF	87
Figura 32: Creación e inserción de registros sobre CARTO.....	87
Figura 33: Información geoespacial sobre CARTO	87
Figura 34: Interfaz inicial Script de importación.....	88
Figura 35: Eliminación de conjunto de datos.....	88



Figura 36: Interfaz inicial Script de importación..... 89

Figura 37: Enumeración de conjunto de datos alojados sobre CARTO 89



Índice de tablas

Tabla 1: Componentes Ontología.....	25
Tabla 2: Resultados Consulta SPARQL.....	26
Tabla 3: Propiedades descriptivas de relaciones geoespaciales	27
Tabla 4: Resultados Consulta Sparql extensión GeoSparql.....	28
Tabla 5: Requerimiento Funcional Código RF001	43
Tabla 6: Requerimiento Funcional Código RF002.....	43
Tabla 7: Requerimiento Funcional Código RF003.....	43
Tabla 8: Requerimiento Funcional Código RF004.....	44
Tabla 9: Requerimiento Funcional Código RF005.....	44
Tabla 10: Requerimiento Funcional Código RF006	44
Tabla 11: Requerimiento Funcional Código RF007	44
Tabla 12: Requerimiento Funcional Código RF008	45
Tabla 13: Requerimiento no Funcional Código RNF001.....	45
Tabla 14: Requerimiento no Funcional Código RNF002.....	45
Tabla 15: Caso de uso Identificador CDU001.....	48
Tabla 16: Caso de uso Identificador CDU002.....	49
Tabla 17: Caso de uso Identificador CDU003.....	49
Tabla 18: Caso de uso Identificador CDU004.....	50
Tabla 19: Caso de uso Identificador CDU005.....	51
Tabla 20: Caso de uso Identificador CDU006.....	51
Tabla 21: Librerías usadas en el script	58
Tabla 22: Script operaciones disponibles	59
Tabla 23: Script Parámetros de configuración.....	59
Tabla 24: Librerías usadas en el Visor.....	63
Tabla 25: Visor operaciones disponibles	64
Tabla 26: Visor parámetros de configuración	65
Tabla 27: Visor parámetros gráficos	69
Tabla 28: Visor opciones de borrado.....	69
Tabla 29: Plataforma Apache Marmotta.....	70
Tabla 30: Perfil de acceso Apache Marmotta	73
Tabla 31: Identificación de parámetros Linkeddata.ec	75
Tabla 32: Visualización de recurso “Provinciasinec”.....	78
Tabla 33: Visualización de recurso “limitecontinental”	78
Tabla 34: Visualización de recurso “proyectogeneracionelectrica”	79
Tabla 35: Visualización de consultas “provincias adyacentes a Pichincha”	80
Tabla 36: Visualización de consultas “Red vial que no cruza por el Azuay”	80



Tabla 37: Visualización de consultas “Centrales eléctricas dentro de Guayas” 81

Tabla 38: Prueba Gestión de parámetros..... 82



Índice de fragmentos de código

Fragmento de código 1: Script archivo de configuración	59
Fragmento de código 2: Prueba de parámetros plataforma CARTO.....	60
Fragmento de código 3: Consulta de selección.....	60
Fragmento de código 4: Código fuente inserción de recursos sobre CARTO.....	61
Fragmento de código 5: Código fuente lista de recursos sobre CARTO.....	61
Fragmento de código 6: Código fuente Elimina recursos sobre CARTO.....	62
Fragmento de código 7: Visor archivo de configuración	64
Fragmento de código 8: Tipo de geometría de recurso	65
Fragmento de código 9: Captación de parámetros gráficos.....	66
Fragmento de código 10: Ejecución consulta hacia CARTO	66
Fragmento de código 11: Captación de consulta preferida por el usuario	67
Fragmento de código 12: Captación de parámetros gráficos.....	67
Fragmento de código 13: Ejecución de consulta hacia Apache Marmotta	68
Fragmento de código 14: Despliegue de recursos sobre el mapa.....	68



Yo *Jonnathan Ramiro Ambrosi Moreira*, autor del Trabajo de Titulación “Visualización y Explotación de información geoespacial en formato RDF (Resource Description Framework)”, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, Abril 2017

Una firma manuscrita en azul que dice "Jonnathan Ambrosi", rodeada por un óvalo azul.

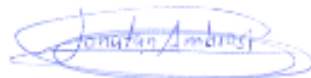
Jonnathan Ramiro Ambrosi Moreira

C.I: 0104613591



Yo *Jonnathan Ramiro Ambrosi Moreira*, autor del Trabajo de Titulación "Visualización y Explotación de información geoespacial en formato RDF (Resource Description Framework)", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor

Cuenca, Abril 2017

Una firma manuscrita en azul que parece decir "Jonnathan Ambrosi", rodeada por un círculo azul.

Jonnathan Ramiro Ambrosi Moreira

C.I: 0104613591



Agradecimientos

A todos aquellos que con su comprensión, amistad y amor supieron acompañarme y guiarme a lo largo de mis estudios, por confortarme y alentarme tanto en los buenos como en los malos momentos. En especial a mis padres que siempre me brindaron su confianza y apoyo a pesar de las dificultades que se presentaron a lo largo del camino recorrido previo a la obtención de este anhelado objetivo.



Dedicatoria

A mi madre que con su infinito amor me enseñó valores como el respeto, amistad y lealtad. A mi padre que a pesar de las dificultades siempre me brindó su apoyo y confianza. Y a todos los que de una u otra forma contribuyeron en mi formación moral y académica.



Capítulo 1

1. Introducción

En el presente capítulo se presenta la definición detallada del proyecto de tesis, en ella se define sus antecedentes, problemática y justificación. Adicionalmente son expuestos tanto el objetivo general como los objetivos específicos. Finalmente se describe la metodología que será usada a lo largo del desarrollo del proyecto, esta ayudara a la obtención y cumplimiento de los objetivos antes mencionados.

1.1. Antecedentes

La implantación de la Web Semántica significa un cambio de paradigma, puesto que propone el cambio de una web sintáctica a una web cuyo principal objetivo es proveer de significado a los datos. Una aplicación de la Web Semántica es Linked Data, que se puede definir como un conjunto de buenas prácticas para publicar y enlazar datos. Con Linked Data tanto personas y máquinas pueden explorar la web de datos enlazados y llegar a información relacionada que se hace referencia desde otros datos iniciales (W3C Definición de Linked Data). De esta forma se pasa de la web basada en documentos HTML¹ con propiedades hipertextuales a una web donde los datos están representados a través de un modelo estándar: RDF² (Resource Description Framework), estos datos en RDF pueden ser explotados mediante procedimientos en función del significado (Castells, 2003).

Actualmente la web está inundada de información de todo tipo, la información de tipo geoespacial no escapa a ello, dicha información está distribuida por la red en formatos heterogéneos y no existe relación alguna entre ella. Al publicar datos enlazados de tipo geoespacial mediante los principios de Linked Data³ se obtiene los datos en RDF, estos datos pueden ser alojados sobre un servidor sobre el cual es posible realizar búsquedas mediante SPARQL⁴. La particularidad de la información geoespacial es que para ser comprendida por el ser humano esta

¹ **HTML:** lenguaje de marcado que se utiliza para el desarrollo de páginas Web (Musciano & Kennedy, 1996).

² **RDF:** modelo de datos estándar que hace posible el intercambio de información sobre la web (Lassila & Swick, 1999).

³ **Linked Data:** término que describe las mejores prácticas recomendadas para publicar, compartir y conectar partes de información y conocimiento sobre la Web semántica (Gómez-Pérez, 2012).

⁴ **SPARQL:** protocolo y lenguaje de consulta sobre grafos RDF propuesto como estándar por el W3C (Hartig, Bizer, & Freytag, 2009).



necesita ser desplegada en forma gráfica (mapas), es por ello que en la fase final de Linked data se realiza la explotación de datos, para lo cual es necesario el desarrollo de herramientas de visualización y explotación de información geoespacial (Villazón-Terrazas & Corcho, 2011).

En el contexto ecuatoriano son varias las fuentes de información de tipo geoespacial, como el Instituto Geográfico Militar (IGM), Programa para el manejo del Agua y el Suelo (PROMAS), la Secretaria Nacional del Agua (SENAGUA), etc. Estos y otros datos geoespaciales han sido publicados en el repositorio ecuatoriano de información de tipo geoespacial GEO LINKED DATA ECUADOR⁵ siguiendo la metodología para publicar Linked Data, es así que a día de hoy se provee acceso público a estos datos geoespaciales ecuatorianos en formato RDF (Novillo, 2016).

1.2. Planteamiento del problema

Actualmente la iniciativa ecuatoriana de datos enlazados de tipo geoespacial GEO LINKED DATA ECUADOR hace uso de la herramienta MAP4RDF⁶ para la visualización y explotación de información de tipo geoespacial, con lenguaje de consulta SPARQL y la capacidad de consultas espaciales con GeoSPARQL⁷. Mediante el uso del estándar GeoSPARQL es posible representar y consultar información geoespacial de diferentes rasgos terrestres como edificios, océanos, ríos, etc. MAP4RDF es una aplicación Web Java que puede ser cargada y desplegada sobre un servidor Web, además permite visualizar fuentes de datos en formato RDF enriquecidas con información geométrica, esta aplicación presenta limitaciones que se describen a continuación.

1. El tiempo es un factor importante en el despliegue y presentación de información geoespacial sobre un mapa, MAP4RDF presenta la información geoespacial en forma correcta sin embargo el tiempo que conlleva este proceso es demasiado largo, por lo que es conveniente reducir el mismo.

⁵ **GEO LINKED DATA ECUADOR:** repositorio ecuatoriano de datos enlazados en el contexto geoespacial.

⁶ **MAP4RDF:** software de código abierto que necesita ser configurado para usar cualquier SPARQL Endpoint y que proporciona a los usuarios una visualización de datos georreferenciados y en RDF en un mapa (Llaves, Corcho, & Fernandez-Carrera, 2014).

⁷ **GeoSPARQL:** estándar establecido por Open Geospatial Consortium (OGC) para la representación de datos geoespaciales en la Web Semántica (Battle & Kolas, 2012).



2. MAP4RDF es una aplicación de código abierto es decir cualquiera que lo desee podrá realizar modificaciones en función de sus necesidades, sin embargo, estas modificaciones están limitadas a la estructura misma de la herramienta, por lo que para realizarlas es necesario contar con un muy alto nivel de programación para conseguir pequeños cambios.
3. La presentación visual de la información geoespacial contenida en RDF es otro factor importante, MAP4RDF no permite gestionar parámetros como color, dimensiones y etiquetas, estos parámetros son constantes y solamente se pueden modificar desde el código fuente.

Las limitaciones antes descritas hacen procedente el uso o desarrollo de otra herramienta de visualización y explotación de información geoespacial de datos enlazados.

1.3. Justificación

Para solucionar los problemas mencionados anteriormente es necesario el desarrollo de una aplicación Web que mejore las prestaciones que ofrece MAP4RDF, esta aplicación será capaz de visualizar y explotar datos enlazados de tipo geoespacial, será implementada sobre el visualizador genérico de información geoespacial de CARTO⁸; que junto con sus API⁹ conforman una poderosa herramienta de visualización. Los usuarios a quienes va dirigida esta aplicación serán capaces de interactuar con ella sin necesidad de tener conocimientos de programación o Linked data. El objetivo de esta aplicación será visualizar y explotar el repositorio que contiene información geoespacial publicada gracias a la iniciativa ecuatoriana de datos enlazados GEO LINKED DATA ECUADOR, además será implantada sobre un servidor del departamento de Ciencias de la computación (DCC) de la Universidad de Cuenca. La propuesta para la explotación de la información geoespacial utiliza como lenguaje de consulta SPARQL y la capacidad de consultas espaciales de GeoSPARQL.

⁸ **CARTO:** plataforma en la nube de código abierto que proporciona tanto el análisis de datos geoespaciales como la visualización de los mismos mediante mapas (González Balea, 2012).

⁹ **API:** es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software

(Fuente: https://es.wikipedia.org/wiki/Interfaz_de_programación_de_aplicaciones).



A continuación se presenta las ventajas que representara el desarrollo de esta aplicación al usar el visualizador genérico de información geoespacial de CARTO.

1. El uso del visualizador genérico de CARTO permite personalizar la aplicación en dirección de los objetivos, también reducir el costo y el tiempo de desarrollo. Esto gracias a que CARTO posee características adaptables a múltiples desarrollos.
2. Al usar CARTO como visualizador de información geoespacial, la gestión de la presentación visual es posible, parámetros como color, dimensiones, etiquetas son gestionables directamente desde la interfaz gráfica de la aplicación.
3. Existe gran cantidad de sitios web que usan CARTO como visualizador de información geoespacial, esto implica gran cantidad de documentación y una comunidad muy extensa de usuarios. La suma de estos factores facilitan el desarrollo en gran medida.
4. Una de las características más importantes al momento de elegir un visualizador de información geoespacial, es el tiempo que le toma en presentar esta información sobre un mapa. CARTO reduce el tiempo de visualización respecto a MAP4RDF en gran medida, esto es gracias a la arquitectura de la plataforma basado en servicios Web.

1.4. Objetivos

1.4.1 Objetivo general

Visualizar y explotar información geoespacial expresada en formato RDF, proveniente del repositorio de GEO LINKED DATA ECUADOR.

1.4.2 Objetivos específicos

- Importar archivos con información geoespacial en formato RDF proveniente de la iniciativa GEO LINKED DATA ECUADOR, hacia la base de datos geoespacial de la plataforma CARTO.



- Visualizar información geoespacial tanto desde CARTO como desde un determinado SPARQL Endpoint¹⁰ (Apache Marmotta¹¹).
- Explotar información geoespacial contenida en el repositorio de GEO LINKED DATA ECUADOR usando lenguaje de consulta SPARQL y su extensión GeoSPARQL.

1.5. Metodología

1.5.1. Búsqueda Bibliográfica

Esta fase básicamente consiste en la búsqueda de artículos científicos, revistas, libros, es decir, toda aquella fuente bibliográfica que pueda proveer información valida referente a los temas de investigación.

1.5.2. Clasificación Bibliográfica

Esta segunda fase consiste en clasificar y filtrar la información recolectada del paso anterior, esto con el fin de desechar todo aquel material que no brinde información relacionada con los temas de investigación.

1.5.3. Investigación Bibliográfica

Con la bibliografía ya clasificada se procede a obtener información y a relacionarla con los distintos puntos que se encuentran en el esquema de investigación.

1.5.4. Redacción

Con la información obtenida en el paso anterior se procederá a redactar el marco teórico del proyecto.

1.5.5. Elección de herramientas

Con los conocimientos adquiridos se realiza una clasificación y comparación de las distintas herramientas existentes para el desarrollo de la aplicación de la presente tesis. Esto implica servidores de tripletas, protocolos, APIs para el manejo de RDF, APIs para la gestión de información geoespacial, lenguajes de consulta, lenguajes

¹⁰ **SPARQL Endpoint:** servicio que permite realizar consultas SPARQL sobre un grafo determinado compuesto por tripletas RDF (Hartig et al., 2009).

¹¹ **Apache Marmotta:** es una plataforma abierta de datos enlazados que puede ser usada, extendida y desplegada fácilmente por organizaciones cuyo objetivo es publicar datos enlazados (Fuente: <https://marmotta.apache.org/>).



de programación, visualizadores genéricos de información geoespacial, IDE¹² de desarrollo, etc.

1.5.6. Diseño

Una vez seleccionada las herramientas más apropiadas se realizan todo el diseño de la aplicación web, esto implicará los modelos y diagramas que representen el funcionamiento, arquitectura, comportamiento e interacción entre los componentes de la aplicación, los mismos que ayudaran a que el proceso de desarrollo se lleve con fluidez.

1.5.7. Implementación

En esta fase se desarrollará la aplicación para la visualización y explotación de datos enlazados de tipo geoespacial.

1.5.8. Pruebas

Esta última fase consistirá en realizar pruebas de la aplicación desarrollada, esto implicará visualizar tanto datos enlazados de tipo geoespacial íntegros como datos enlazados producto de una consulta geoespacial hacia un Sparql Endpoint.

1.5.9. Implantación

En esta fase la aplicación desarrollada será desplegada sobre un servidor perteneciente al departamento de ciencias de la computación (DCC) de la Universidad de Cuenca.

¹² **IDE:** es una aplicación informática que proporciona servicios integrales para facilitar al desarrollador o programador el desarrollo de software
(Fuente: https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado).

Capítulo 2

2. Conceptos previos

En este capítulo se describen términos referentes a metodologías, herramientas y plataformas necesarias para cumplir con los objetivos planteados en la presente tesis. Es por ello que se realizó una elección basada en las características que brindan estas herramientas y en las necesidades requeridas por el proyecto, es así que se optó por Apache Marmotta como plataforma de almacenamiento de RDF con información geoespacial, para realizar la explotación de estos datos enlazados geoespaciales se construyó una aplicación Web que toma como base el visualizador genérico de CARTO, además el uso de varias herramientas de desarrollo como Apache Jena, CartoDB-Java-Client, Carto.js, Carto.css y GeoTools.

2.1. Web semántica

La web semántica es un área pujante en la confluencia de la Inteligencia Artificial y las tecnologías web que propone introducir descripciones explícitas sobre el significado de los recursos, para permitir que las propias máquinas tengan un nivel de comprensión de la web suficiente como para hacerse cargo de una parte, la más costosa, rutinaria, o físicamente inabarcable, del trabajo que actualmente realizan manualmente los usuarios que navegan e interactúan con la web (Castells, 2003).

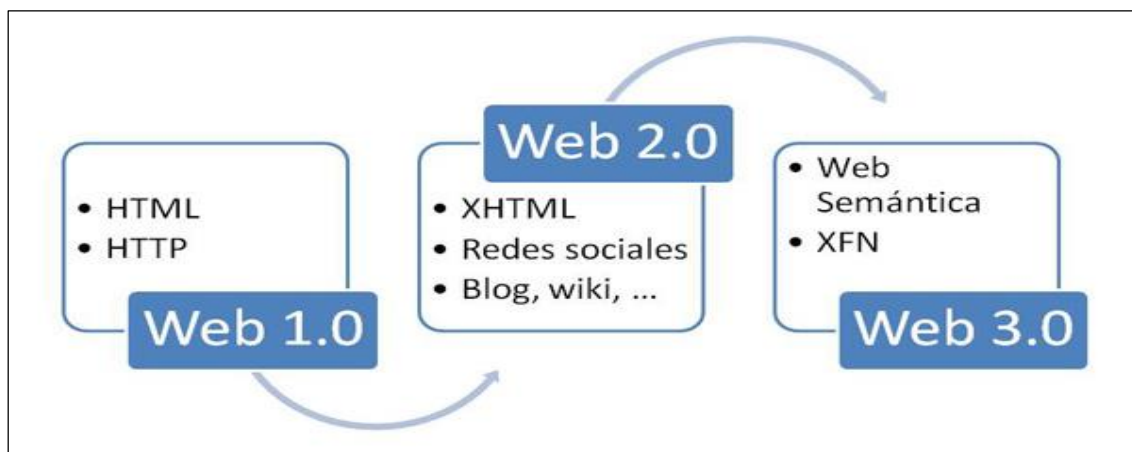


Figura 1: Evolución de la Web (fuente: La web Social, <http://e-educativa.catedu.es>)

2.1.1. Definición de web semántica

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web

de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante (Montero, Viedma, Torres, & Redondo, 2003).

2.1.2. RDF (Resource Description framework)

Es un modelo estándar que sirve para el intercambio de datos en la Web. RDF tiene características que facilitan la mezcla de datos incluso si sus esquemas son diferentes, además soporta la evolución de esquemas en el tiempo.

RDF utiliza tripletas que están compuestas por un sujeto, un predicado y un objeto para representar recursos en forma de grafos. Estos grafos pueden ser procesados por herramientas informáticas. RDF es capaz de representar información de cualquier entidad en forma de grafo, la información geoespacial no escapa a ello en la Figura 2.2 se describe gráficamente la relación entre un recurso su nombre y su geometría (Tello, 2001).

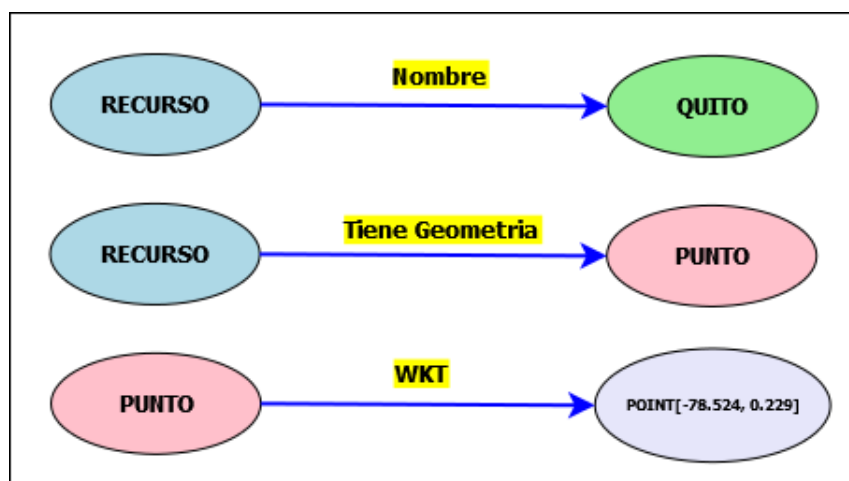


Figura 2: Representación recurso Quito

2.1.3. Ontología

La ontología es un concepto filosófico (parte de la metafísica que trata el ser en general y de sus propiedades trascendentales), adoptado por la informática. En su aplicación a la Web Semántica, alude al intento de formular un exhaustivo y riguroso mapa conceptual dentro de uno o varios dominios dados, con la finalidad



de facilitar la comunicación e intercambio de información (interoperabilidad) entre varios sistemas.

Las ontologías definen de forma estándar y consensuada un vocabulario de conceptos así como las relaciones entre ellos dentro de un área concreta del conocimiento, formando redes jerárquicas semánticas. Así recogen reglas lógicas y restricciones para hacer “comprender” a las máquinas los conceptos que manejan dentro de un determinado campo. Ejemplo; Una ontología de arte describe que todos los escultores son artistas pero no todos los artistas son escultores. En la tabla 1 se describe una ontología en función de sus componentes (Tello, 2001).

Componentes	Descripción
Conceptos	Son las clases que se intentan formalizar dentro de un dominio determinado.
Instancias	Representan objetos determinados de un concepto.
Relaciones	Representan la interacción y enlace entre conceptos del dominio.
Funciones	Tipo concreto de relación.
Axiomas	Son teoremas que aplican la lógica en las relaciones

Tabla 1: Componentes Ontología

2.1.4. SPARQL y GeoSPARQL

A continuación se describe las características del lenguaje de consulta SPARQL y su extensión GeoSPARQL.

SPARQL

Es un conjunto de especificaciones que provee lenguajes y protocolos para consultar y manipular grafos RDF que están contenidos sobre la Web o sobre un triple store. Es una tecnología clave en el desarrollo de la Web Semántica que se constituyó como Recomendación oficial del W3C el 15 de enero de 2008 (Battle & Kolas, 2012).

En la figura 3 se visualiza un ejemplo de una consulta Sparql hacia el triple store Apache Marmotta. El objetivo de esta consulta es listar los nombres de las provincias del Ecuador y sus geometrías en formato WKT.

```
PREFIX geontology: <http://geo.linkeddata.ec/ontology/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?label_respuesta ?wkt_respuesta
FROM <http://190.15.132.241/marmotta/context/geoecuator>
WHERE {
  ?subject2 a <http://geo.linkeddata.ec/ontology/provincias_promsa>.
  ?subject2 rdfs:label ?label_respuesta.
  ?subject2 geontology:hasExactGeometry ?geo2.
  ?geo2 geo:asWKT ?wkt_respuesta.
}
```

Figura 3: Sintaxis Consulta SPARQL

En la tabla 2 se visualiza el resultado producto de la consulta descrita en la Figura 3.

Nombre	Geometría Formato WKT
CHIMBORAZO@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])
BOLIVAR@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])
GUAYAS@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])
⋮	⋮
AZUAY@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])

Tabla 2: Resultados Consulta SPARQL

GeoSPARQL

Es un estándar establecido por Open Geospatial Consortium (OGC) definido para la representación y acceso a datos geoespaciales en el contexto de la Web Semántica. Para cumplir su cometido provee una representación común de datos geoespaciales en formato RDF, además posee la capacidad de gestionar las relaciones existentes entre las entidades geoespaciales (Battle & Kolas, 2012).

En la tabla 3 se describen las propiedades para definir las relaciones geoespaciales definidas en el estándar GeoSparql.

Propiedad	Descripción
ogc:equals	Igualad entre dos entidades geoespaciales
ogc:disjoint	Intersección geoespacial no existente entre dos entidades geoespaciales
ogc:intersects	Intersección geoespacial existente entre dos entidades geoespaciales
ogc:touches	Coincidencia de al menos un punto de su periferia, y el interior no es intersectado.
ogc:within	La entidad geoespacial se encuentra totalmente dentro de otra entidad geoespacial
ogc:contains	La entidad geoespacial contiene totalmente a otra entidad geoespacial
ogc:overlaps	Las entidades geoespaciales comparten parcialmente un mismo espacio.

Tabla 3: Propiedades descriptivas de relaciones geoespaciales

En la figura 4 se visualiza un ejemplo de una consulta Sparql que hace uso de la extensión GeoSPARQL y su propiedad sfcontains, esta es direccionada hacia el triple store Apache Marmotta que soporta dicho estándar. El objetivo de esta consulta es obtener el nombre y la geometría en formato WKT¹³ de las lagunas que se encuentran dentro de la provincia de Azuay.

```
PREFIX geontology: <http://geo.linkeddata.ec/ontology/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?label_respuesta ?wkt_respuesta
FROM <http://190.15.132.241/marmotta/context/geoecuator>
WHERE {
    ?subject a <http://geo.linkeddata.ec/ontology/provincias_promsa>.
    ?subject rdfs:label "AZUAY"@es.
    ?subject geontology:hasExactGeometry ?geo.
    ?geo geo:asWKT ?wkt.
    ?subject2 a <http://geo.linkeddata.ec/ontology/lagunas>.
    ?subject2 rdfs:label ?label_respuesta.
    ?subject2 geontology:hasExactGeometry ?geo2.
    ?geo2 geo:asWKT ?wkt_respuesta.
    FILTER (geof:sfContains(?wkt, ?wkt_respuesta))
}
```

Figura 4: Sintaxis Consulta SPARQL extensión GeoSPARQL

¹³ **WKT**: La representación Well Known Text o de texto conocido es una codificación o sintaxis en formato ASCII estandarizada diseñada para describir objetos espaciales expresados de forma vectorial (Fuente: https://es.wikipedia.org/wiki/Well_Known_Text).

A continuación en la tabla 4 se visualiza el resultado producto de la consulta descrita en la figura 4.

Nombre	Geometría Formato WKT
LAGARTOCOCHA@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])
OSOHUAYCU@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])
CASCARILLAS@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])
:	:
VENTANAS@es	<http://www.opengis.net/def/crs/OGC/1.3/CRS84> MULTIPOLYGON ([lat, lon], [lat, lon], ..., [lat, lon])

Tabla 4: Resultados Consulta Sparql extensión GeoSparql

2.2. Datos enlazados y datos enlazados geoespaciales

A continuación se describe tanto datos enlazados como datos enlazados en el contexto geoespacial. En la figura 5 se puede observar un extracto de la Nube de Datos enlazados en donde se encuentra visible la iniciativa ecuatoriana de datos enlazados geoespaciales.

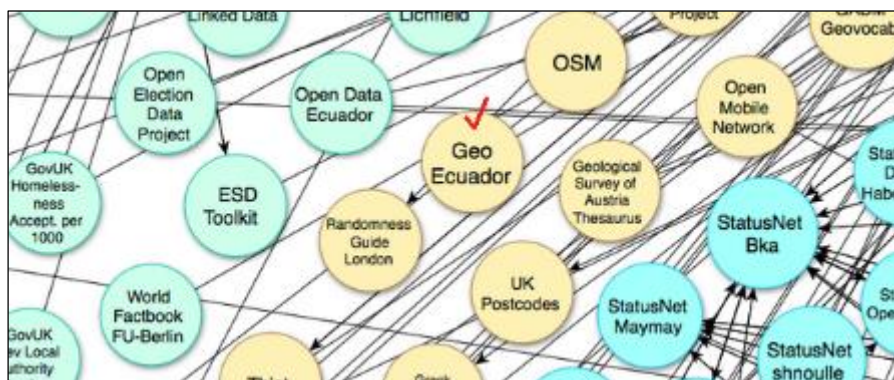


Figura 5: Extracto Nube de Datos Enlazados (fuente: Linked Open Data Diagram, <http://lod-cloud.net/>)

2.2.1. Datos enlazados

Los datos enlazados es la forma que tiene la web semántica de entrelazar datos que se encuentran distribuidos sobre la Web, la Web semántica no se trata únicamente de la publicación de datos en la Web, si no que estos pueden relacionar a otros, de forma que las personas y las maquinas puedan explorar la web de los datos.



Los Datos Enlazados, como parte de la Web Semántica, se basa en la aplicación de ciertos principios básicos y necesarios, que fomentarán el crecimiento de la Web, tanto a nivel de los documentos HTML (vista clásica de la Web), como a nivel de los datos expresados en RDF (vista de la Web Semántica) (Bizer, Heath, & Berners-Lee, 2009).

- Utilizar URIs¹⁴ para identificar los recursos.
- Utilizar el protocolo HTTP¹⁵ para nombrar y resolver la ubicación de los datos identificados mediante esas URIs.
- Representar los datos en RDF y utilizar Sparql como lenguaje de consulta de dichos datos.
- Incluir enlaces a otros URIs para permitir la localización de más datos enlazados.

2.2.2. Datos enlazados geoespaciales

Los datos enlazados geoespaciales es la forma que tiene la web semántica de entrelazar datos geoespaciales que se encuentran distribuidos sobre la Web. Una vez que los datos son publicados es posible realizar la visualización y explotación de los mismos mediante aplicaciones como MAP4RDF, Sexant, SemMap y lenguajes de consulta como SPARQL.

2.3. Metodología para la generación y publicación de datos enlazados geoespaciales

A continuación se describen cada una de las fases que comprenden la metodología para la publicación de datos enlazados en el contexto geoespacial. Esta metodología propone un modelo cíclico incremental basado en las mejores continuas de los datos enlazados generados, esta metodología comprende las fases especificación, modelado, generación de RDF, generación de Links, publicación y explotación.

¹⁴ **URI:** es una cadena de caracteres que identifica los recursos de una red de forma unívoca (Fuente: https://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme).

¹⁵ **HTTP:** es el protocolo de comunicación que permite las transferencias de información en la World Wide Web (Fuente: https://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol).

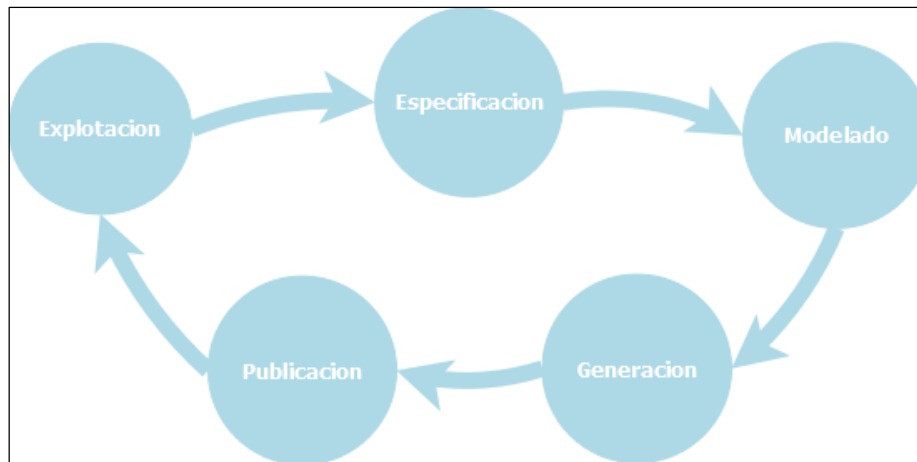


Figura 6: Fases Metodología generación de Datos enlazados

2.3.1. Especificación

El primer objetivo de esta fase de la metodología de datos enlazados geoespaciales, es identificar la localización y el formato en que se encuentran estos datos, además es necesario analizarlos en términos de confiabilidad y validez de ello dependerá en gran medida la validez de los datos enlazados generados.

El segundo objetivo de esta fase es diseñar el formato o patrón de las URIs que serán útiles para la posterior identificación de los recursos, el diseño de estas URIs es relevante en la metodología, ya que éstas contribuirán de manera clave en el alineamiento de instancias provenientes de diferentes fuentes de información (Vilches-Blázquez, Sevilla, Villalón, Rodríguez, & Gómez-Pérez, n.d.).

2.3.2. Modelado

El objetivo de esta fase es definir una ontología que sea capaz de representar los datos contenidos en las fuentes de información definidas en la fase de especificación. En caso de no existir la adecuada es necesario modificar una o varias existentes y en último caso la creación de otra que se adapte a las necesidades requeridas por las fuentes de información y sus datos.

2.3.3. Generación de RDF

Una vez definidos las fuentes de información y la ontología para su representación el siguiente paso es generar archivos en formato RDF que contengan estos datos, para ello es necesario el uso de herramientas como Geometry2RDF¹⁶, cuyo objetivo

¹⁶ **Geometry2RDF:** es una librería para generar ficheros RDF a partir de información geométrica que puede estar disponible en GML o WKT (Fuente: mayor2.dia.fi.upm.es/oeg-upm/index.php/es/technologies/151-geometry2rdf/).



es generar archivos en formato RDF tomando como entrada múltiples formatos de fuentes de datos y un archivo de configuración (Gómez-Pérez, 2012).

A continuación se presentan los parámetros del archivo de configuración

- Directorio de trabajo y nombre del fichero de salida.
- Namespaces y prefijos para los recursos generados y para la ontología usada.
- Sistema de referencia.

A continuación se presentan características de Geometry2RDF

- Trabaja con bases de datos Geoespaciales.
- Soporta formato de salida WKT y GML¹⁷.
- Requiera Java 1.5 o superior.

2.3.4. Generación de Links

El objetivo de esta actividad es la generación de relaciones también llamadas enlaces entre los datos enlazados geoespaciales generados en formato RDF con diferentes conjuntos de datos de la Web de Datos enlazados geoespaciales.

Para cumplir con este objetivo se utilizan aplicaciones como Silk Workbench, la cual es una aplicación Web que a través de un proceso permite gestionar una o varias condiciones de enlazado para encontrar recursos relacionados entre sí, de esta manera los datos son enriquecidos y ampliados mediante la existencia de enlaces (Volz, Bizer, Gaedke, & Kobilarov, 2009).

Silk Workbench provee las siguientes características.

- Permite al usuario gestionar diferentes conjuntos de información, tareas de enlace y transformación.
- Provee al usuario un entorno gráfico que facilita las tareas de enlace y transformación.
- Generalmente la búsqueda de una buena heurística de enlace suele ser un proceso iterativo, esta aplicación hace posible que el usuario evalúe rápidamente los enlaces que son generados por la especificación de enlace actual.

¹⁷ **GML:** Lenguaje de Marcado Geográfico que está destinado al modelaje, transporte y almacenamiento de información geográfica (Fuente: <https://es.wikipedia.org/wiki/GML>).



2.3.5. Publicación

El objetivo de esta fase es permitir el acceso a estos datos en formato RDF mediante el uso de lenguajes de consulta como SPARQL. Es por ello que es necesario cargar estos archivos RDF antes generados en un triple store o servidor de tripletas RDF, estos permiten la gestión de estos archivos. A continuación se lista varios servidores de tripletas RDF.

- OpenLink Virtuoso
- Apache Marmotta
- Parliament
- Fuseki

2.3.6. Explotación

Finalmente una vez que los archivos en formato RDF están cargados en un servidor de tripletas, el acceso o consumo de ellos es posible mediante un SPARQL Endpoint que es un servicio que ofrecen los servidores de tripletas siendo necesario el lenguaje de consulta Sparql como nexo entre las aplicaciones externas y el servidor contenedor de la información.

Plataformas de soporte de explotación de RDF con información geoespacial.

- MAP4RDF
- Sextant
- SemMap
- LINKEDGEODATABROWSER

Al tratar con información en el contexto geoespacial es necesario que las aplicaciones que consumen estos datos los desplieguen sobre mapas, con ello esta información es interpretada por el ser humano de forma clara. Existen varias aplicaciones que cumplen con este objetivo sin embargo también es posible la implementación de aplicaciones a medida dependiendo de las necesidades presentes, para ello es común el uso de visualizadores genéricos que son altamente acoplables a múltiples desarrollos (González Balea, 2012).

Visualizadores genéricos de información geoespacial

- Google Fusion Tables
- GeoCommons
- Carto

En la figura 7 se observa el despliegue de información geoespacial ecuatoriana en archivos en formato RDF sobre un mapa.

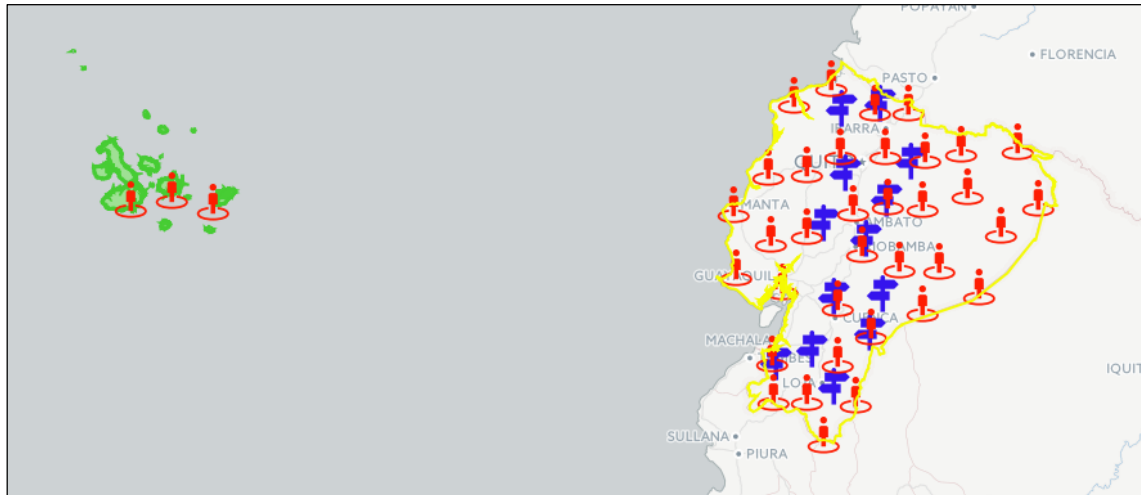


Figura 7: Despliegue de información geoespacial sobre un mapa

2.4. Plataformas de soporte de almacenamiento de RDF con información geoespacial

En esta sección se describe varias herramientas que permiten la carga y almacenamiento de archivos en formato RDF, que han sido generados siguiendo la metodología de datos enlazados.

2.4.1. OpenLink Virtuoso

Virtuoso Universal Server es un software informático que proporciona servicios a las aplicaciones de software más allá de los disponibles en el sistema operativo y también un motor de base de datos que combina la funcionalidad de RDBMS¹⁸ tradicional, ORDBMS¹⁹, base de datos virtual, RDF, XML²⁰, texto libre, servidor de aplicaciones en un solo sistema (Provider, 2009).

¹⁸ **RDBMS:** Sistema de gestión de bases de datos relacionales, es un programa que permite crear, actualizar y administrar una base de datos relacional (Fuente: https://es.wikipedia.org/wiki/Base_de_datos).

¹⁹ **ORDBMS:** es una extensión de la base de dato relacional tradicional, a la cual se le proporcionan características de la programación orientada a objetos (Fuente: https://es.wikipedia.org/wiki/Base_de_datos_objeto-relacional).

²⁰ **XML:** es un meta lenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible (Fuente: https://es.wikipedia.org/wiki/Extensible_Markup_Language).



A continuación se listan las prestaciones ofrecidas por esta herramienta

- Gestión de datos relacionales
- Gestión de datos RDF
- Gestión de datos XML
- Servidor web de documentos
- Servidor de datos enlazados

2.4.2. Apache Marmotta

Apache Marmotta es una plataforma abierta de datos enlazados que puede ser usada, extendida y desplegada fácilmente por organizaciones cuyo objetivo es publicar datos enlazados o construir aplicaciones personalizadas de datos enlazados.

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Permite lectura y escritura de datos enlazados
- Almacenamiento de tripletas RDF y razonamiento basado en reglas
- Lenguajes de consulta LDPPath y SPARQL
- Mecanismos de seguridad integrados

2.4.3. Parliament

Parliament es un servidor de tripletas de alto rendimiento diseñado para la Web Semántica, originalmente fue desarrollado bajo el nombre DAML DB y fue extendido por BBN Technologies para uso interno en R&D programs. Parliament fue relanzado como proyecto de código abierto bajo licencia BSD en Junio de 2009 (Battle & Kolas, 2012).

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Se encuentra en desarrollo constante
- Incluye soporte SPARQL
- Incluye soporte de persistencia tanto Jena como Sesame
- Incluye indexación y procesamiento geoespacial

2.4.4. Fuseki

Fuseki es un servidor de tripletas RDF desarrollado por Apache Jena, para su acceso se utilizan servicios web de tipo REST²¹ utilizando el protocolo HTTP, y

²¹ **REST:** es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web (Fuente: https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional).



como lenguaje de consulta usa SPARQL, además brinda interfaces para los siguientes estándares (J. Jena & Fuseki, 2004).

- SPARQL Query
- SPARQL Update
- SPARQL Protocol
- SPARQL Graph Store HTTP Protocol

2.5. Plataformas de explotación de información geoespacial que soportan RDF y GeoSPARQL

En esta sección se describe varias herramientas Web que permiten la explotación de información geoespacial contenida en archivos RDF, que se encuentran cargados en un servidor de tripletas RDF

2.5.1. Map4RDF

Map4RDF es un software de código abierto que simplemente necesita ser configurado para usar cualquier SPARQL Endpoint y que proporciona a los usuarios una visualización de datos georreferenciados y en RDF en un mapa. Los aspectos geoespaciales de los datos se pueden modelar usando o el modelo de datos del W3C Geo XG o GeoSPARQL (Leon, Wisniewski, Villazón-Terrazas, & Corcho, 2012).

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Visualización geométrica y geoespacial usando Google Maps y Open Street Maps.
- Filtrado de consultas.
- Visualización de geometrías GeoSPARQL formato WKT.
- Configuración vía panel de administración o archivo de configuración.

2.5.2. Sextant

Sextant es una plataforma web para visualizar, explorar e interactuar con datos enlazados geoespaciales. En un inicio era necesario que el usuario final tenga un alto nivel de conocimiento de Sparql para poder interactuar con ella. Actualmente está implementada de tal forma que sea posible interactuar con ella tanto para usuarios expertos en el dominio de datos enlazados como para usuarios no expertos (Nikolaou, Dogani, Kyzirakos, & Koubarakis, 2013).



A continuación se listan las prestaciones ofrecidas por esta herramienta

- Creación de mapas temáticos a partir de varios SPARQL Endpoint.
- Soporte estándar SPARQL y su extensión GeoSPARQL.
- Soporte para crear, compartir y editar mapas.
- Soporte para exportar mapas hacia formatos comunes compatibles con otros GIS²².

2.5.3. SemMap

Esta aplicación permite al usuario explorar el contenido de un Sparql Endpoint de manera gráfica. Permite al usuario seleccionar una porción específica de datos que le interesa, creando una "faceta" del conjunto de datos. La faceta se crea mediante la definición de un conjunto de restricciones sobre las propiedades de la base de datos. Una vez que se define la faceta, se puede hacer clic en la información de la faceta en una tabla y visualizarla en un mapa (Mader, Martin, & Stadler, 2014).

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Funciona para conjuntos de datos grandes: Si una región contiene demasiados elementos, se requiere que el usuario aumente el zoom para obtener una visualización correcta.
- Al estar desarrollado íntegramente con Java Script es de fácil integración con varios sitios Web.
- Solo es necesaria la configuración de un Sparql Endpoint.

2.5.4. LINKEDGEODATABROWSER

LGD Browser and Editor permite visualizar recursos terrestres usando un mapa mundial. Es un esfuerzo para agregar una dimensión espacial a la web de los datos también denominada web semántica, para ello usa los datos recolectados por el proyecto OpenStreetMap además estos datos estén disponibles en formato RDF siguiendo los principios de la metodología Linked data (Auer, Lehmann, & Hellmann, 2009).

²² **GIS:** es un conjunto de herramientas que integra y relaciona diversos componentes (usuarios, hardware, software, procesos) que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial

(Fuente: https://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n_geogr%C3%A1fica).



A continuación se listan las prestaciones ofrecidas por esta herramienta

- Usa Open Link Virtuoso como servidor de tripletas RDF
- Provee servicio de SPARQL Endpoint
- Los datos son enriquecidos mediante relaciones con fuentes de información como DBpedia y Geonames
- Al tratarse de una iniciativa mundial el proyecto está disponible en varios idiomas

2.6. Visualizadores genéricos de información geoespacial

En esta sección se describe varias herramientas que permiten visualización de información geoespacial contenida en múltiples formatos, proveen APIs que permiten crear aplicaciones que son alojadas en sitios web para la exploración de información en el contexto geoespacial.

2.6.1. Google Fusion Tables

Es un servicio web de Google para la gestión de datos, los datos se almacenan en tablas que los usuarios pueden gestionar, además este sitio web proporciona un medio para visualizar los datos con gráficos circulares, barras, diagramas de dispersión, líneas de tiempo, así como mapas geográficos basados en Google Maps. Los datos pueden exportarse como archivos de texto y los valores son separados por comas (Halevy & Shapley, 2009).

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Permite alojar los datos en la nube.
- Crear visualizaciones personalizadas para sus datos.
- Analizar los datos alojados en las tablas.
- Control de acceso a las tablas que contienen los datos.
- Combinación de tablas.

2.6.2. GeoCommons

GeoCommons es una plataforma geoespacial de gestión de datos, visualización y análisis. La interfaz gráfica de la aplicación es muy intuitiva, además es posible gestionar parámetros gráficos como color, tamaño, transparencia, etc. (Macdonald, 2008).



A continuación se listan las prestaciones ofrecidas por esta herramienta

- Permite el trabajo Colaborativo
- Orientado a usuarios técnicos y no técnicos.
- La comunidad de usuarios es muy amplia.
- Posee varios mapas base tales como topográfico, callejero, satélite y otros.
- Soporta múltiples operaciones entre entidades geoespaciales como Intersection, merge, addition, buffer, etc.

2.6.3. Carto

Carto es una plataforma web que permite la visualización de información geoespacial, esta plataforma es de código abierto sin embargo también ofrece soporte para cuentas bajo pago dependiendo de las necesidades de las entidades que la requieren. Esta plataforma para su almacenamiento usa una base de datos PostgreSQL²³ con extensión Postgis²⁴ lo que le permite gestionar la información de tipo geoespacial de manera eficiente, para crear visualizaciones personalizadas esta provista de APIs como CartoDB.js y CartoDB.css.

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Gracias a Carto.css es posible personalizar las visualizaciones de manera rápida y eficaz.
- Gracias a Carto.js y su arquitectura basada en servicios web es posible crear sitios web personalizados.
- Existen múltiples formas de importar la información entre ellas sql, url e interfaz gráfica.
- Existe una gran comunidad lo que facilita el desarrollo.
- Es posible definir la privacidad de los datos tanto como públicos o privados.

2.7. Herramientas de desarrollo y soporte

En esta sección se describen varias herramientas que permiten gestionar la transmisión de información geoespacial tanto desde el servidor de tripletas RDF

²³ **PostgreSQL:** es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT
(Fuente: <https://es.wikipedia.org/wiki/PostgreSQL>).

²⁴ **Postgis:** es una extensión que convierte el sistema de base de datos PostgreSQL en una base de datos espacial. La combinación de ambos es una solución perfecta para el almacenamiento, gestión y mantenimiento de datos espaciales
(Fuente: <https://mappinggis.com/2012/09/por-que-utilizar-postgis/>).



como desde CARTO hacia la aplicación web, su tratamiento y finalmente su visualización.

2.7.1. Apache Jena

Jena es un framework Java que sirve para construir aplicaciones de la Web semántica, también provee una variedad de librerías Java que permiten gestionar RDF, RDFS²⁵, RDFa²⁶, OWL²⁷ y Sparql, además incluye un motor de inferencia basado en reglas que se ejecuta sobre ontologías OWL y RDFS. Jena se desarrolló en HP Labs en el 2000, en 2009 HP cedió el proyecto a la fundación Apache que decidió adoptarlo en noviembre de 2010 (A. Jena, 2007).

A continuación se listan las prestaciones ofrecidas por esta herramienta

- API para lectura, escritura y procesamiento de ontologías RDF y OWL
- Motor de inferencia para razonar sobre ontologías RDF y OWL
- Almacenamiento de tripletas RDF flexible es decir tanto en disco como en memoria.
- Motor de consultas compatible con Sparql.

2.7.2. GeoTools

GeoTools es una biblioteca SIG de código abierto que permite desarrollar soluciones estandarizadas, proporciona una implementación de las especificaciones del Open Geospatial Consortium según estas van apareciendo (Turton, 2008).

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Está escrito en lenguaje de programación Java
- Se encuentra en desarrollo continuo.
- La comunidad de usuarios es muy extensa lo que facilita el desarrollo.
- Muchos SIG hacen uso de ella por su concepción modular.

²⁵ **RDFS:** es una extensión semántica de RDF, un lenguaje primitivo de ontologías que proporciona los elementos básicos para la descripción de vocabularios
(Fuente: https://es.wikipedia.org/wiki/RDF_Schema).

²⁶ **RDFa:** es un conjunto de extensiones de XHTML propuestas por W3C para introducir semántica en los documentos. RDFa aprovecha atributos de los elementos meta y link de XHTML y los generaliza de forma que puedan ser utilizados en otros elementos
(Fuente: <https://es.wikipedia.org/wiki/RDFa>).

²⁷ **OWL:** un lenguaje de marcado para publicar y compartir datos usando ontologías en la WWW. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML (Fuente: <https://es.wikipedia.org/wiki/OWL>).



- Un renderizador sin estado, de bajo consumo de memoria, particularmente útil en entornos de ejecución de servidor
- Soporte de transacciones y bloqueo entre hilos de ejecución.

2.7.3. CartoDB-Java-Client

Es una librería Java que permite acceder a la base de datos geoespacial CARTO mediante el SQL API que provee la plataforma, para ello son necesarios dos parámetros como son el nombre de usuario y una clave de aplicación, ambos proporcionados por la plataforma en el momento en que se solicita y se crea una cuenta.

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Usa SQL como lenguaje de consulta enriquecido con funciones geoespaciales.
- El formato de respuesta a las solicitudes HTTP es JSON²⁸.
- Se encuentra disponible en varios lenguajes de programación en el repositorio GitHub.
- Es posible construir un archivo en formato jar a partir de un proyecto Gradle²⁹.
- Existe abundante documentación sobre el proyecto.

2.7.4. Carto.js

Es una librería JavaScript que permite interactuar con los servicios de la plataforma CARTO mediante el uso de la Web. Esta librería permite conectarnos hacia nuestros conjuntos de datos geoespaciales que residen en la plataforma, crear visualizaciones de nuestros datos, personalizar nuestras visualizaciones, ejecutar consultas desde el navegador, es decir provee a nuestro sitio Web la capacidad de gestionar información geoespacial de maneja visual.

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Existe una gran cantidad de Webs que hacen uso de esta herramienta.
- Existe una gran comunidad lo que facilita el desarrollo.

²⁸ **JSON:** es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo
(Fuente: www.json.org/json-es.html).

²⁹ **Gradle:** es básicamente una herramienta para automatizar el proceso de construcción de nuestro proyecto es decir tareas como compilar, pruebas y empaquetado
(Fuente: www.arquitecturajava.com/que-es-gradle/).



- Con pocas líneas de código se pueden crear vistosas visualizaciones.
- Fue desarrollado por Leaflet³⁰ como plugin para la plataforma CARTO.
- La versión más reciente es la 3.15.8

2.7.5. Carto.css

Es una hoja de estilo que permite personalizar nuestras visualizaciones, es decir permite gestionar varios parámetros tales como color, opacidad, ancho, color de línea, color de relleno, etiquetas, tipo de letra, marcadores, etc. El uso conjunto de esta hoja de estilo con Carto.js permite que las visualizaciones de nuestros conjuntos de datos sean llamativas a la vista.

A continuación se listan las prestaciones ofrecidas por esta herramienta

- Provee el uso de múltiples propiedades aplicables a distintos tipos de geometrías como puntos líneas y polígonos.
- Es posible personalizar el uso de imágenes en formato svg y png como marcadores.
- El código está disponible para el uso o modificación del mismo.

³⁰ **Leaflet:** es una librería JavaScript opensource para crear mapas interactivos en un entorno móvil (Fuente: <https://mappinggis.com/2013/06/como-crear-un-mapa-con-leaflet/>).



Capítulo 3

3. Análisis y diseño de la solución

En el presente capítulo se presenta la definición formal de los modelos, diagramas y especificaciones, que permitan resolver la problemática definida en el capítulo 1, para ello es necesario desarrollar todas las actividades relacionadas con el análisis y diseño de la solución utilizando la metodología UWE (UML Web Engineering). El éxito o fracaso de la solución a implementar depende en gran medida de estas actividades.

3.1. Análisis

El objetivo de esta sección es definir un conjunto de tareas y actividades que permitan cubrir las necesidades de la aplicación que se pretende desarrollar, dichas tareas y actividades deben ser detalladas de manera clara, precisa.

3.1.1. Especificaciones

Las especificaciones comprenden todas las funciones que la aplicación será capaz de realizar, mediante la interacción con el usuario final, a continuación se presentan dichas funciones.

- Visualización sobre el mapa de recursos íntegros alojados en CARTO.
- Visualización sobre el mapa de recursos producto de una consulta SPARQL ejecutada sobre Apache Marmotta.
- Visualización de consultas en lenguaje de consulta SPARQL.
- Gestión de parámetros gráficos como color, transparencia, ancho e imagen de marcador.
- Borrado de recursos desplegados sobre el mapa.

3.1.2. Requerimientos funcionales

Un requerimiento funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Los requerimientos de comportamiento para cada requerimiento funcional se muestran en los casos de uso. Son complementados por los requerimientos no funcionales, que se enfocan en el diseño.



Para definir los requerimientos se usaron los siguientes identificadores en forma tabular.

- Código: Su función es identificar el requerimiento.
- Nombre: Su función es nombrar el requerimiento.
- Descripción: Su función es describir de forma general el requerimiento.
- Prioridad: Su función es identificar la importancia dentro de la aplicación.

Código	RF001
Nombre	Listar Nombre de Recursos
Descripción	Al iniciar la aplicación Web, se lista los nombres de los recursos alojados en el servidor CARTO
Prioridad	Alta

Tabla 5: Requerimiento Funcional Código RF001

Código	RF002
Nombre	Listar Consultas
Descripción	Al iniciar la aplicación Web, se lista las consultas precargadas en la aplicación Web
Prioridad	Alta

Tabla 6: Requerimiento Funcional Código RF002

Código	RF003
Nombre	Cargar mapa base
Descripción	Al iniciar la aplicación Web, se carga el mapa base de la aplicación en donde se desplegaran gráficamente los recursos
Prioridad	Media

Tabla 7: Requerimiento Funcional Código RF003



Código	RF004
Nombre	Desplegar gráficamente recursos sobre el mapa
Descripción	Al seleccionar la casilla correspondiente al recurso se despliega el recurso sobre el mapa
Prioridad	Alta

Tabla 8: Requerimiento Funcional Código RF004

Código	RF005
Nombre	Desplegar gráficamente recursos productos de una consulta SPARQL sobre el mapa.
Descripción	Al pulsar el botón “Ejecutar” correspondiente a cada consulta el recurso producto de la consulta SPARQL es desplegado sobre el mapa
Prioridad	Alta

Tabla 9: Requerimiento Funcional Código RF005

Código	RF006
Nombre	Visualizar Consulta en lenguaje de consulta SPARQL
Descripción	Al pulsar el botón “Consulta” correspondiente a cada consulta, se despliega la consulta el lenguaje SPARQL.
Prioridad	Baja

Tabla 10: Requerimiento Funcional Código RF006

Código	RF007
Nombre	Gestión de parámetros gráficos
Descripción	El usuario tiene la opción de gestionar parámetros como color, ancho, transparencia e imagen de marcador según sus preferencias
Prioridad	Media

Tabla 11: Requerimiento Funcional Código RF007



Código	RF008
Nombre	Remover recursos desplegados sobre el mapa.
Descripción	Al pulsar los botones “Borrar Recursos” correspondientes a recursos o consultas los recursos son removidos del mapa instantáneamente.
Prioridad	Alta

Tabla 12: Requerimiento Funcional Código RF008

3.1.3. Requerimientos no funcionales

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento.

Se definirán los requerimientos no funcionales usando la representación usada en el punto anterior.

Código	RNF001
Nombre	Interfaz sencilla para el usuario
Descripción	La interfaz de la aplicación debe ser amigable con el usuario es decir que pueda ser usada en su totalidad sin complicación alguna
Prioridad	Alta

Tabla 13: Requerimiento no Funcional Código RNF001

Código	RNF002
Nombre	Adaptable a cualquier navegador moderno
Descripción	La interfaz de la aplicación debe ser desplegada de igual manera en cualquier navegador moderno
Prioridad	Alta

Tabla 14: Requerimiento no Funcional Código RNF002



3.1.4. Definición de los límites del sistema

La aplicación web permitirá visualizar información geoespacial contenida en RDF, el origen de esta información es tanto el servidor CARTO como el servidor Apache Marmotta. Además la visualización será complementada mediante la gestión de parámetros gráficos como color, ancho, transparencia e imagen de marcador. También esta aplicación será desplegada sobre el servidor de LinkedData.ec perteneciente al Departamento de Ciencias de la Computación de la Universidad de Cuenca, esta será capaz de interactuar completamente con usuarios que no están familiarizados con los términos como Datos Enlazados o lenguajes de programación.

3.1.5. Lenguaje de programación

Para el desarrollo de la aplicación se utilizara el lenguaje de programación Java. Algunas de las características más importantes por las que se ha elegido este lenguaje son las siguientes:

- Es orientado a objetos. Por lo tanto, al igual que otros lenguajes orientados a objetos, tiene la ventaja de que facilita la reutilización y la extensión del código.
- Es multiplataforma, es decir, un programa escrito en Java se pueda ejecutar en múltiples plataformas.
- Evita muchas preocupaciones al programador. La administración de la memoria es automática, el programador no tiene que ocuparse de la liberación de memoria reservada dinámicamente, ya que un recolector de basura libera la memoria asignada a un objeto cuando ya no exista ninguna referencia a ese objeto.

Al ser una aplicación web es necesario el uso de lenguajes de marcado como HTML para la presentación de contenidos, para la personalización de los mismos es necesario el uso de hoja de estilos CSS, para consumir servicios Web es necesario el uso de AJAX³¹ y JQUERY³².

³¹ **AJAX:** es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página (Fuente: <https://es.wikipedia.org/wiki/AJAX>).

³² **JQUERY:** es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web (Fuente: <https://es.wikipedia.org/wiki/JQuery>).

3.2. Diseño

El objetivo de esta sección es definir el comportamiento y la presentación de cada uno de los elementos que componen la aplicación Web, al ser una aplicación Web dinámica es decir que los contenidos son variables en el tiempo, es necesario modelar cada uno de estos cambios para que esta funcione de manera correcta.

3.2.1. Diagrama casos de uso

El diagrama de casos de uso representa la forma en como un usuario (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan. Un diagrama de casos de uso consta de los siguientes elementos.

- Actor
- Casos de uso
- Relaciones

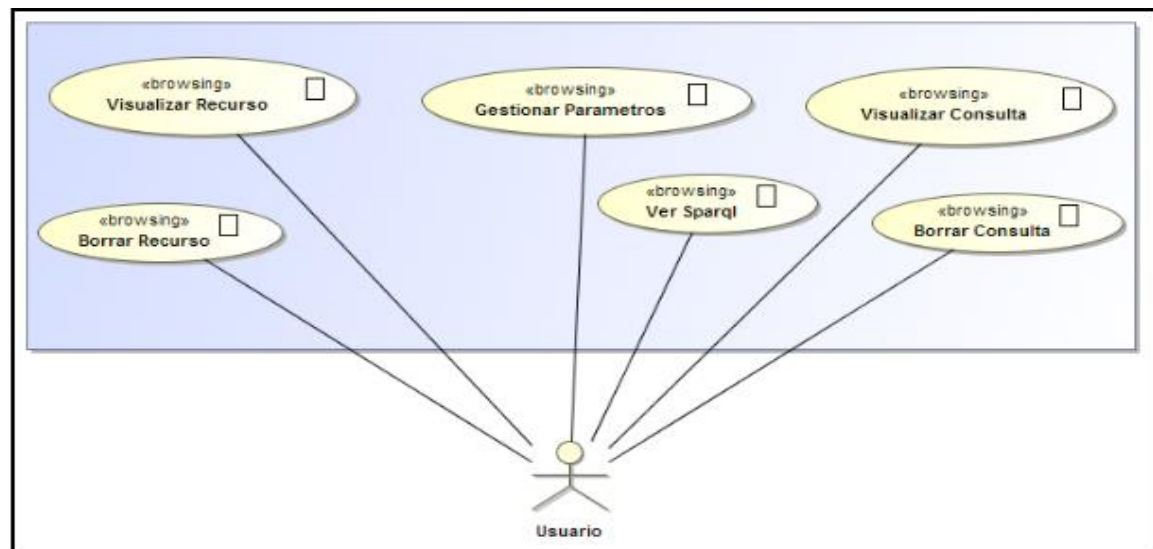


Figura 8: Diagrama de casos de uso

3.2.2. Casos de uso en formato extendido

Un caso de uso en formato extendido es un documento narrativo que describe el comportamiento de un sistema desde el punto de vista de un actor. El actor es una entidad externa del sistema que participa en el caso de uso, suelen ser seres humanos o cualquier tipo de sistema.

El formato que utilizaremos, emplea los siguientes campos:

- Identificador: Cadena que identifica caso de uso
- Caso de Uso: Nombre del caso de uso.
- Objetivo: Explicación del caso del objetivo del caso de uso



- Actor principal: En este caso el Usuario
- Precondiciones: Condiciones que deben cumplirse antes de comenzar el escenario principal de éxito.
- Escenario principal de éxito: Pasos detallados de las interacciones entre actores.
- Garantías de éxito: Qué debe cumplirse cuando el caso de uso se acaba con éxito.
- Garantías de fracaso: Qué debe cumplirse cuando el caso de uso se abandona.

Identificador	CDU001
Caso de uso	Visualizar Recurso
Objetivo	Desplegar gráficamente sobre el mapa un recurso
Actor principal	Usuario
Precondiciones	El recurso debe estar precargado sobre CARTO, además disponer de conexión a internet.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario se posiciona sobre la opción del menú “Recursos” 2. El sistema despliega los Recursos disponibles, además una casilla de selección junto a cada uno de ellos. 3. El usuario selecciona la casilla perteneciente al recurso que quiere visualizar. 4. El sistema ejecuta un procedimiento tomando como parámetro el nombre del recurso. 5. El sistema bloquea la aplicación mientras el recurso es desplegado sobre el mapa.
Garantías de éxito	El recurso fue desplegado sobre el mapa.
Garantías de fracaso	Mensaje de Error.

Tabla 15: Caso de uso Identificador CDU001

Identificador	CDU002
Caso de uso	Visualizar Consulta
Objetivo	Desplegar gráficamente sobre el mapa un recurso producto de una consulta SPARQL
Actor principal	Usuario
Precondiciones	Los recursos involucrados en la consulta deben estar precargados en Marmotta, además disponer de conexión a internet.



Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario se posiciona sobre la opción del menú “Consultas” 2. El sistema despliega las Consultas disponibles, además un botón “Ejecutar” y un botón “Consulta” junto a cada uno de ellas 3. El usuario presiona el botón “Ejecutar” correspondiente a la consulta que quiere visualizar 4. El sistema ejecuta un procedimiento tomando como parámetro la consulta SPARQL 5. El sistema bloquea la aplicación mientras la consulta es desplegada sobre el mapa
Garantías de éxito	La consulta fue desplegada sobre el mapa.
Garantías de fracaso	Mensaje de Error.

Tabla 16: Caso de uso Identificador CDU002

Identificador	CDU003
Caso de uso	Ver Consulta en Lenguaje SPARQL
Objetivo	Desplegar una consulta SPARQL sobre la interfaz
Actor principal	Usuario
Precondiciones	La consulta debe estar precargada en el sistema
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario se posiciona sobre la opción del menú “Consultas” 2. El sistema despliega las Consultas disponibles, además un botón “Ejecutar” y un botón “Consulta” junto a cada uno de ellas 3. El usuario presiona el botón “Consulta” correspondiente a la consulta que quiere visualizar 4. El sistema ejecuta un procedimiento tomando como parámetro título de la consulta 5. El sistema muestra la consulta SPARQL sobre la interfaz.
Garantías de éxito	La consulta en SPARQL fue desplegada sobre la interfaz
Garantías de fracaso	Mensaje de Error.

Tabla 17: Caso de uso Identificador CDU003



Identificador	CDU004
Caso de uso	Borrar Recursos
Objetivo	Borrar Recursos desplegados sobre el mapa
Actor principal	Usuario
Precondiciones	Existencia de al menos un recurso que se encuentre desplegado sobre el mapa.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario se posiciona sobre la opción del menú “Opciones de Borrado” 2. El sistema despliega dos botones junto con su descripción 3. El usuario pulsa el botón correspondiente a borrar recursos 4. El sistema ejecuta un procedimiento tomando como parámetro la descripción del botón pulsado 5. Todos los recursos desplegados sobre el mapa son removidos
Garantías de éxito	No existe ningún recurso desplegado sobre el mapa
Garantías de fracaso	El estado anterior al caso de uso en el que estaba el sistema

Tabla 18: Caso de uso Identificador CDU004

Identificador	CDU005
Caso de uso	Borrar Consultas
Objetivo	Borrar Consultas desplegados sobre el mapa
Actor principal	Usuario
Precondiciones	Existencia de al menos una consulta que se encuentre desplegada sobre el mapa.
Escenario de éxito	<ol style="list-style-type: none"> 1. El usuario se posiciona sobre la opción del menú “Opciones de Borrado” 2. El sistema despliega dos botones junto con su descripción 3. El usuario pulsa el botón correspondiente a borrar consultas 4. El sistema ejecuta un procedimiento tomando como parámetro la descripción del botón pulsado 5. Todas las consultas desplegadas sobre el mapa son removidas
Garantías de éxito	No existe ninguna consulta desplegado sobre el mapa



Garantías de fracaso	El estado anterior al caso de uso en el que estaba el sistema

Tabla 19: Caso de uso Identificador CDU005

Identificador	CDU006
Caso de uso	Gestionar parámetros gráficos
Objetivo	Definir parámetros gráficos
Actor principal	Usuario
Precondiciones	No existen
Escenario de éxito	<p>1 El usuario se posiciona sobre la opción del menú “Gestión de parámetros gráficos”</p> <p>2 El sistema despliega varios elementos que permiten definir color, transparencia, ancho e imagen como marcador</p> <p>3 El usuario define los valores de los parámetros según sus preferencias graficas</p> <p>4 El sistema guarda sus preferencias en componentes HTML</p>
Garantías de éxito	Los recursos y las consultas graficadas son desplegadas con los parámetros establecidos
Garantías de fracaso	El estado anterior al caso de uso en el que estaba el sistema

Tabla 20: Caso de uso Identificador CDU006

3.2.3. Diagrama de clases

En ingeniería de software, un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

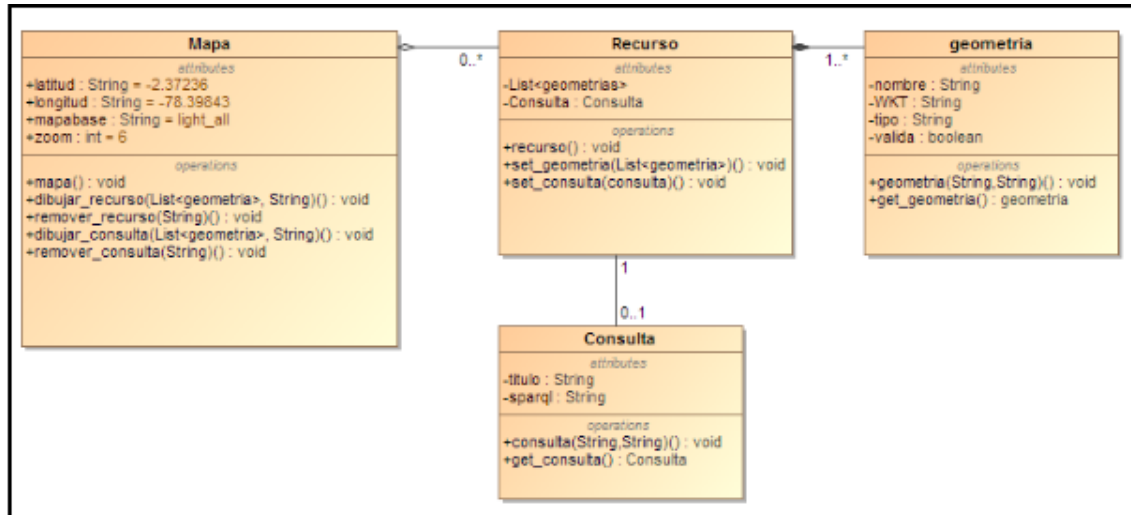


Figura 9: Diagrama de clases

3.2.4. Arquitectura del sistema

La arquitectura de un sistema es una representación de un sistema existente o a crear, y el proceso y disciplina para efectivamente implementar el diseño como un sistema. Este sistema se compone de tres herramientas a continuación se describe a detalle su participación dentro del sistema.

- **Plataforma Apache Marmotta:** Esta plataforma contendrá la información geoespacial ecuatoriana en archivos en formato RDF, además la plataforma provee de un servicio SPARQL Endpoint el cual será atacado por la aplicación de visualización desarrollada. Es necesario aclarar que la plataforma será desplegada y configurada sobre el servidor *linkeddata.ec*.
- **Plataforma CARTO:** La plataforma provee de un servicio de base de datos geoespacial, es por ello que contendrá la información geoespacial ecuatoriana. Para consumir esta información desde la aplicación de visualización desarrollada la plataforma provee APIs que permiten la gestión tanto de la base de datos así como de la presentación visual de esta información.

- **Visor de información geoespacial:** Esta aplicación Web es la encargada de consumir y desplegar de forma gráfica sobre mapas la información geoespacial proveniente de la iniciativa ecuatoriana de datos enlazados *linkeddata.ec*. Esta es capaz de interactuar tanto con la plataforma Apache Marmotta así como con la plataforma CARTO. Es necesario aclarar que la aplicación será desplegada y configurada sobre el servidor *linkeddata.ec*.

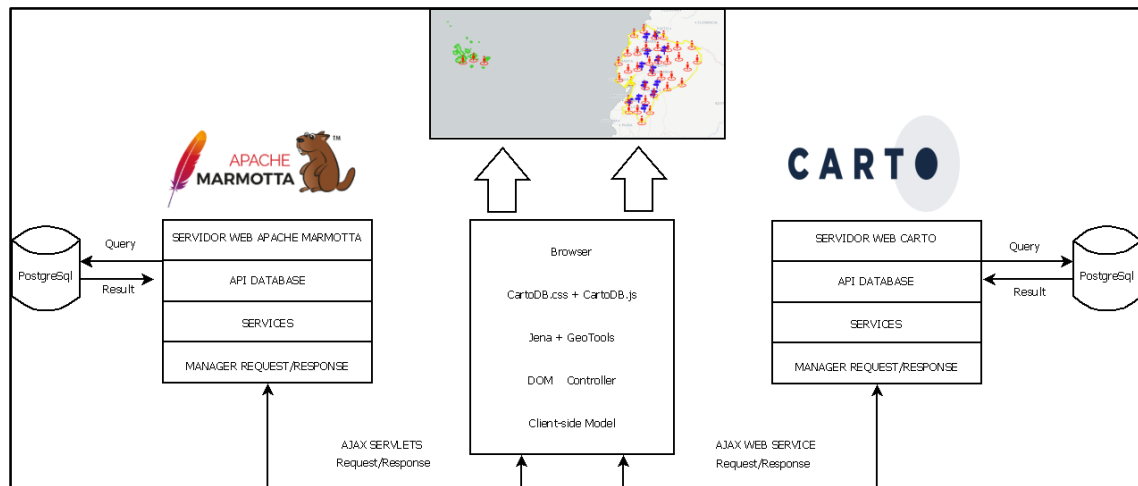


Figura 10: Arquitectura Visor de información geoespacial

3.2.5. Diseño de interfaz

El diseño de la interfaz de usuario es uno de los procesos más importantes dentro del proceso de desarrollo, ya que permite la comunicación entre el ser humano y el computador. La interfaz de usuario es la encargada de transmitir todas las peticiones del ser humano y también se encarga de desplegar los resultados generados por la aplicación. Es importante hacer notar que la interfaz de usuario no es responsable de ejecutar procesos o almacenar información.

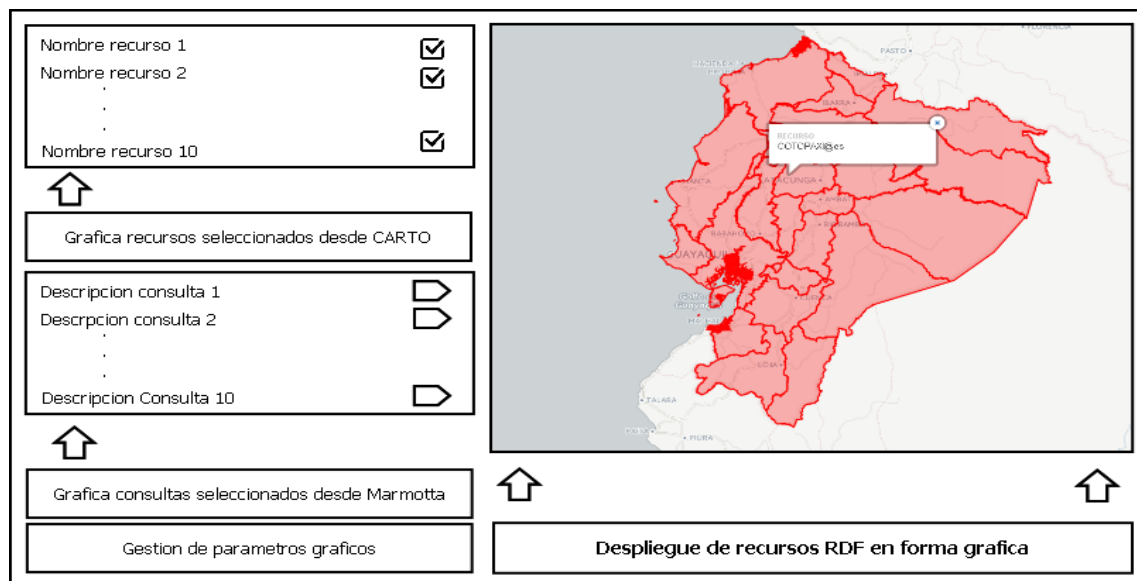


Figura 11: Interfaz Visor de información geoespacial

Capítulo 4

4. Implementación de la solución

En el presente capítulo se describen las aplicaciones desarrolladas que de forma conjunta permiten visualizar la información geoespacial contenida en archivos en formato RDF proveniente de la iniciativa ecuatoriana de datos enlazados linkeddata.ec, también se describen las herramientas y plataformas usadas que permiten la gestión de esta información entre ellas la base de datos geoespacial de la plataforma CARTO y el servidor de tripletas RDF Apache Marmotta.

4.1. Descripción de componentes de la solución

A continuación se presenta información de cada uno de los componentes de la solución, así como extractos de código fuente de las aplicaciones desarrolladas.

4.1.1. Descripción Base de datos geoespacial CARTO

CARTO es una plataforma geoespacial web que provee múltiples servicios tanto gratuitos como pagos, entre los más importantes están un servicio de base de datos PostgreSQL con extensión Postgis, un conjunto de APIs que permiten gestionar la base de datos, para el control de seguridad provee de una clave de aplicación que permite acceso seguro y para la comunicación con aplicaciones usa servicios web que son invisibles para el desarrollador.

cartodb_id number	the_geom geometry	recurso string	
1	Polygon	TULCAN@es	
2	Polygon	ESPEJO@es	
3	Polygon	MONTUFAR@es	

Figura 12: Estructura base de datos plataforma CARTO

Actualmente el número de archivos en formato RDF que contienen información geoespacial que han sido alojados en la base de datos de la plataforma CARTO asciende a 80, estos archivos provienen de la iniciativa ecuatoriana de datos enlazados linkeddata.ec, la información contenida en estos archivos describen rasgos terrestres del Ecuador tales como cantones, provincias, ríos, centros educativos, parques nacionales, etc.

4.1.2. Descripción Servidor de tripletas RDF Apache Marmotta

Apache Marmotta es un servidor de tripletas RDF de código abierto el cual está en desarrollo constante, específicamente en este proyecto permite la gestión de archivos en formato RDF que contienen información geoespacial proveniente de la iniciativa linkeddata.ec, además la versión construida usa el lenguaje de consulta Sparql y la capacidad de consultas geoespaciales mediante el estándar GeoSPARQL, también provee un SPARQL Endpoint el cual permite consumir datos desde la aplicación de visualización desarrollada.

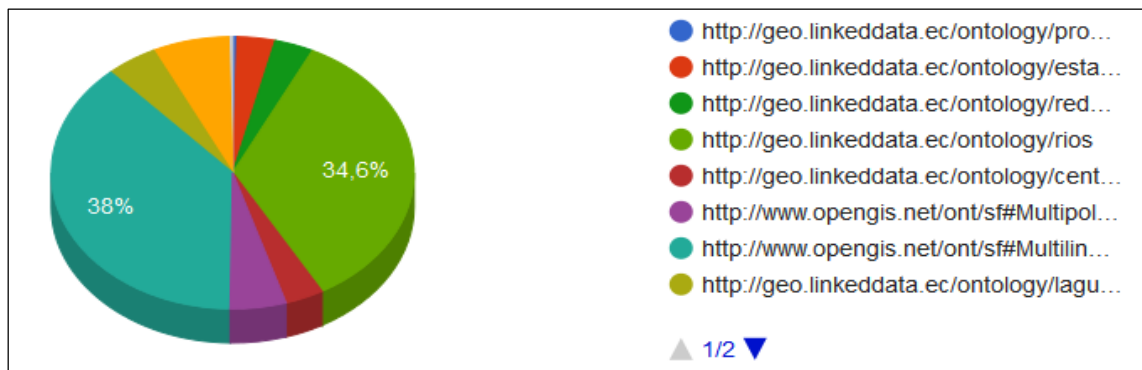


Figura 13: Extracto grafos almacenados sobre Apache Marmotta

Actualmente el número de archivos en formato RDF que contienen información geoespacial que han alojados en la base de datos del servidor de tripletas RDF Apache Marmotta asciende a 80, la información contenida en estos archivos describen rasgos terrestres del ecuador tales como cantones, provincias, ríos, centros educativos, parques nacionales, etc.

4.1.3. Descripción Script para importar RDF hacia CARTO

El objetivo que persigue el desarrollo de este script es importar los archivos en formato RDF con información geoespacial proveniente de la iniciativa linkeddata.ec hacia la base de datos geoespacial de la plataforma CARTO, una vez alojados allí es posible consumirlos mediante APIs desde la aplicación de visualización desarrollada.

```

+-----+
SCRIPT IMPORTAR RDF GEOESPACIAL HACIA CARTO
+-----+
-----LISTA DE OPCIONES-----

A) Ver Usuario y Api Key de Acceso ha CARTO.
B) Ver Consultas SPARQL de Seleccion.
C) Probar parametros Usuario y Api Key.
D) Probar Consultas SPARQL de Seleccion.
E) Ver nombre de DataSet de CARTO.
F) Eliminar por nombre de DataSet de CARTO.
G) Importar DataSet RDF ha CARTO.
S) Salir.
Digite una opcion..
    
```

Figura 14: Interfaz y operaciones de Script

4.1.4. Descripción Visor de información geoespacial.

El objetivo del visor desarrollado es consumir información geoespacial, tanto desde la base de datos geoespacial de la plataforma CARTO como desde el servidor de tripletas RDF Apache Marmotta, una vez que la información está disponible en la aplicación esta es desplegada gráficamente sobre el mapa de esta manera es comprensible totalmente por el usuario.

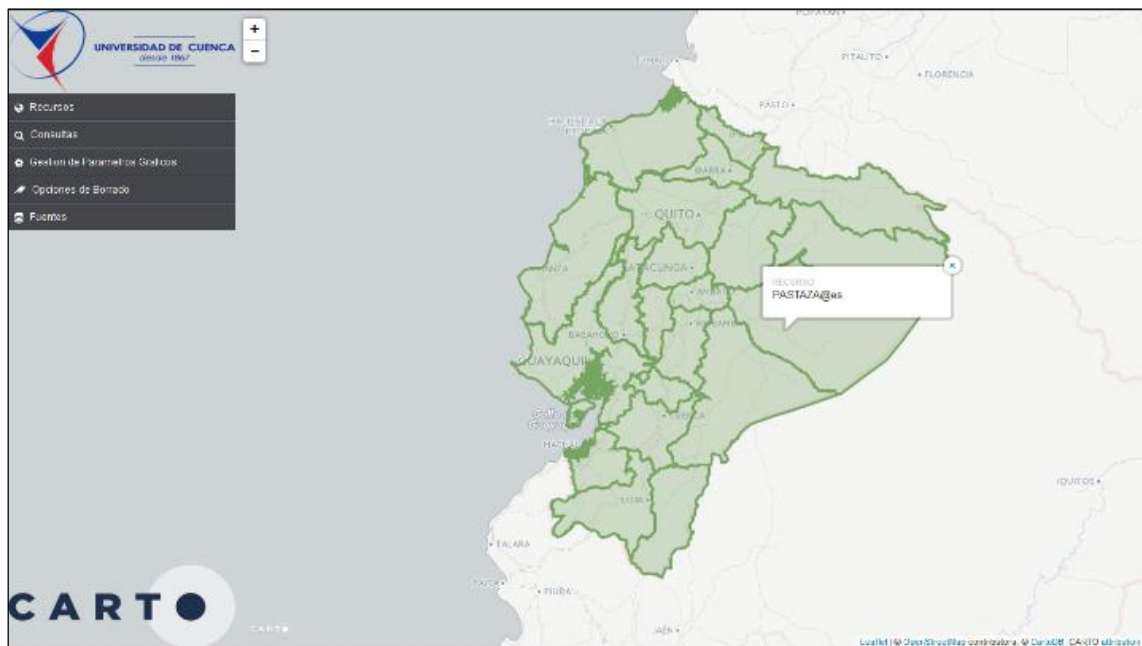


Figura 15: Visor de información geoespacial contenida en RDF



4.2. Script para importar RDF hacia CARTO

En esta sección se describe el proceso de desarrollo, funcionalidades, librerías necesarias, configuración de parámetros, lenguaje de programación y entorno de desarrollo involucrados en la construcción de este script.

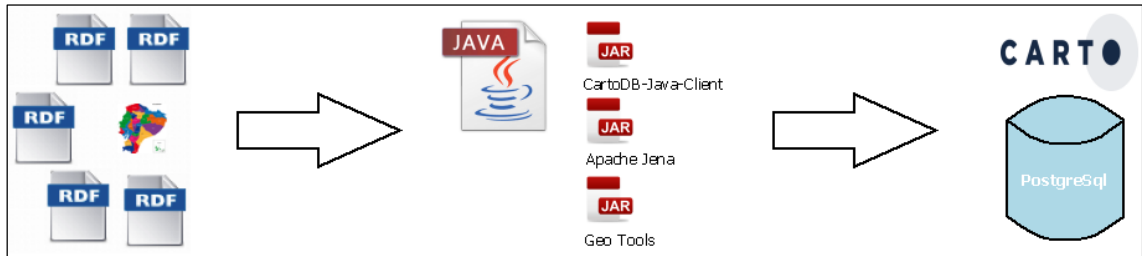


Figura 16: Arquitectura Script para importar RDF hacia CARTO

4.2.1. Entorno y lenguaje de desarrollo

Como entorno de desarrollo se optó por el IDE Netbeans por su facilidad de uso y el conocimiento sobre esta herramienta, como lenguaje de programación se optó por Java por su interoperabilidad entre sistemas y su sencillez.

4.2.2. Librerías necesarias

Al tratarse de un script que gestiona información almacenada en formato RDF, además de conectarse con la base de datos geoespacial de la plataforma CARTO y gestionar información geoespacial en formato WKT es necesario contar con las siguientes librerías Apache Jena, GeoTools y el API CartoDB-java-Client que provee la plataforma CARTO.

Librería	Web
Apache Jena	https://jena.apache.org/
GeoTools	http://www.geotools.org/
CartoDB-java-Client	https://github.com/CartoDB/cartodb-java-client

Tabla 21: Librerías usadas en el script

4.2.3. Funcionalidades

La tabla que se muestra a continuación describe todas las operaciones que es capaz de ejecutar el script junto con sus opciones.

Opción	Descripción
A	Sirve para conocer los parámetros obtenidos desde el archivo de configuración, cuyo objetivo es el acceso a los servicios de la plataforma CARTO



B	Despliega textualmente en la consola las consultas de selección ingresadas en la aplicación.
C	Nos indica si los parámetros de acceso a los servicios de la plataforma CARTO son correctos, es decir si existe una cuenta con tales parámetros.
D	Nos indica si las consultas de selección producen resultados al ejecutarla sobre un RDF, es decir si el conjunto de datos resultante posee filas (nombre de recurso, geometría).
E	Lista todas las tablas correspondientes al usuario con las credenciales ingresadas en el archivo de configuración.
F	Elimina determinado conjunto de datos como parámetro es necesario el nombre del conjunto de datos.
G	Una vez se prueban las consultas de selección se procede a cargar los archivos RDF hacia la plataforma CARTO.

Tabla 22: Script operaciones disponibles

4.2.4. Parámetros de configuración

El objetivo de este archivo de configuración es centralizar la información necesaria para el correcto funcionamiento del script.

```
# Parametros Plataforma CARTO
usuario=tomb88
api_key=b9c60645a6d011a2e8aacc40835d0404cf0728d8
# Contienen el resultado de la consulta de seleccion
var_label=label_respuesta
var_wkt=wkt_respuesta
```

Fragmento de código 1: Script archivo de configuración

Parámetro	Descripción
Usuario	Contiene el nombre de usuario de la plataforma CARTO
Clave de aplicación	Contiene la clave de aplicación de la plataforma CARTO
var_label	Contiene el nombre del recurso tras la consulta
var_wkt	Contiene el WKT del recurso tras la consulta

Tabla 23: Script Parámetros de configuración



4.2.5. Prueba de parámetros de acceso a la plataforma CARTO

El objetivo de esta opción es validar las credenciales de acceso a la plataforma CARTO. Para ello se realiza una simple consulta, si esta consulta produce resultados quiere decir que los parámetros son correctos caso contrario son inválidos.

```
System.out.println("+-----+");
System.out.println("+-----PRUEBA DE PARAMETROS CARTO-----+");
System.out.println("+-----+");

Properties prop = new Properties();
prop.load(getClass().getResourceAsStream("/config/parametros.properties"));
String user=prop.getProperty("usuario");
String api_key=prop.getProperty("api_key");

CartoDBClientIF cartoDBClient=new ApiKeyCartoDBClient(user,api_key);
String consulta="select relname from pg_stat_user_tables WHERE schemaname='public'";
String cadena=cartoDBClient.executeQuery(consulta);
if(cadena.contains("total_rows"))
{
    System.out.println("Parametros Correctos");
}
else
{
    System.out.println("Parametros Incorrectos");
}
```

Fragmento de código 2: Prueba de parámetros plataforma CARTO

4.2.6. Consultas de selección

El objetivo de las consultas de selección es obtener los nombres de los recursos así como su geometría en formato WKT, para su posterior inserción en la base de datos de la plataforma CARTO.

```
mis_consultas.add(
new consulta(
"PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>\n" +
"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\n" +
"PREFIX time: <http://www.w3.org/2006/time#>\n" +
"PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>\n" +
"PREFIX geontology: <http://geo.linkeddata.ec/ontology/>\n" +
"PREFIX geo: <http://www.opengis.net/ont/geosparql#>\n" +
"PREFIX sf: <http://www.opengis.net/ont/sf#>\n" +
"SELECT ?label_respuesta ?wkt_respuesta \n" +
"WHERE \n" +
"{" +
" ?subject rdfs:label ?label_respuesta.\n" +
" ?subject geontology:hasExactGeometry ?geo.\n" +
" ?geo geo:asWKT ?wkt_respuesta.\n" +
"}", "Selecciona el nombre y geometria de todos lo recursos"));
```

Fragmento de código 3: Consulta de selección

4.2.7. Importar RDF con información geoespacial hacia CARTO

Una vez que los parámetros de acceso han sido validados y que las consultas de selección han producido resultados sobre el archivo en formato RDF que se requiere alojar sobre la base de datos, se procede a seleccionarlo e insertarlo sobre la base de datos de la plataforma CARTO.

```
Properties prop = new Properties();
prop.load(getClass().getResourceAsStream("/config/parametros.properties"));
String user=prop.getProperty("usuario");
String api_key=prop.getProperty("api_key");

CartoDBClientIF cartoDBClient=new ApiKeyCartoDBClient(user,api_key);
String tipo=mr.get(0).getTipo();
//CREAR TABLA
sql="CREATE TABLE IF NOT EXISTS "+nombre_tabla+" ("+"tipo+" geometry,recurso varchar)";
resul=cartoDBClient.executeQuery(sql);
System.out.println("Estado creacion de la tabla :"+resul+"\n");
//INSERCIÓN DE REGISTROS
String wkt_geometria;
String recurso_nombre;
for(int i=0;i<mr.size();i++)
{
    if(mr.get(i).isValida())
    {
        recurso_nombre=mr.get(i).getRecurso();
        wkt_geometria=mr.get(i).getWKT();
        sql="INSERT INTO "+nombre_tabla+" VALUES ( '"+wkt_geometria+"', '"+recurso_nombre+"' ) ";
        resul=cartoDBClient.executeQuery(sql);
        System.out.println("Insercion numero "+(i+1)+" :"+resul);
    }
}
//CARTODIFICAR LA TABLA
sql="select cdb_cartodbifytable('"+nombre_tabla+"')";
resul=cartoDBClient.executeQuery(sql);
System.out.println("Visible en 2 minutos: "+resul);
System.out.println("Pulse INTRO para continuar..");
br.readLine();
```

Fragmento de código 4: Código fuente inserción de recursos sobre CARTO

4.2.8. Listar Recursos alojados sobre CARTO

El objetivo de esta opción es listar los nombres de los conjuntos de datos que han sido importados hacia la base de datos de la plataforma CARTO.

```
System.out.println("+-----+");
System.out.println("+-----LISTA DE CONSULTAS DE SELECCION-----+");
System.out.println("+-----+");
for(int i=0;i<mc.get_size_list();i++)
{
    System.out.println(mc.get_title_by_index(i));
    System.out.println(mc.get_query_by_index(i));
    System.out.println("Pulse cualquier tecla para continuar..");
    br.readLine();
}
```

Fragmento de código 5: Código fuente lista de recursos sobre CARTO

4.2.9. Eliminar Recursos alojados sobre CARTO

El objetivo de esta opción es eliminar los conjuntos de datos que han sido importados hacia la base de datos de la plataforma CARTO, como parámetro es necesario el nombre de conjunto de datos.

```
System.out.println("+-----+");
System.out.println("+-----ELIMINAR DATASET CARTO-----+");
System.out.println("+-----+");

System.out.println("Ingrese el nombre de dataset a eliminar : ");
String dataset_name=br.readLine();

Properties prop = new Properties();
prop.load(getClass().getResourceAsStream("/config/parametros.properties"));
String user=prop.getProperty("usuario");
String api_key=prop.getProperty("api_key");

CartoDBClientIF cartoDBClient=new ApiKeyCartoDBClient(user,api_key);
String cadena=cartoDBClient.executeQuery("DROP TABLE "+dataset_name+" ");
if(cadena.contains("total_rows"))
{
    System.out.println("Dataset eliminado correctamente");
}
else
{
    System.out.println("Error");
}
```

Fragmento de código 6: Código fuente Elimina recursos sobre CARTO



4.3. Visor de información geoespacial

En esta sección se describe el proceso de desarrollo, funcionalidades, librerías necesarias, configuración de parámetros, lenguaje de programación y entorno de desarrollo involucrados en la construcción de la aplicación para visualización de información geoespacial.

4.3.1. Entorno y lenguaje de desarrollo

Como entorno de desarrollo se optó por el IDE Netbeans, además como lenguaje de programación en el lado del servidor se optó por Java y en el lado del cliente se optó por JavaScript. Estas herramientas permitieron la construcción de aplicación de visualización de información geoespacial, fueron seleccionadas por su facilidad de uso y el conocimiento que existe sobre ellas.

4.3.2. Librerías necesarias

Al tratarse de una aplicación web que despliega gráficamente información geoespacial sobre un mapa, es necesario contar con las siguientes librerías Apache Jena, GeoTools, JQuery, MarkerCluster.js y las APIs CartoDB-java-Client, Carto.js, Carto.css que provee la plataforma CARTO. De esta manera es posible gestionar información en formato RDF, gestionar información en formato WKT, conectarse con la base de datos geoespacial de la plataforma CARTO y la base de datos del servidor de tripletas RDF Apache Marmotta.

Librería	Web
Apache Jena	https://jena.apache.org/
GeoTools	http://www.geotools.org/
CartoDB-java-Client	https://github.com/CartoDB/cartodb-java-client
Carto.js	http://libs.cartocdn.com/cartodb.js/v3/3.15/cartodb.js
Carto.css	http://libs.cartocdn.com/cartodb.js/v3/3.15/themes/css/cartodb.css
MarkerCluster	http://asset.geosprocket.com/leaflet/cluster/leaflet.markercluster.js
Jquery	https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js

Tabla 24: Librerías usadas en el Visor



4.3.3. Funcionalidades

La tabla que se muestra a continuación describe todas las operaciones que es capaz de ejecutar la aplicación web de visualización de información geoespacial.

Opción	Descripción
Recursos	El objetivo de esta opción es desplegar gráficamente sobre un mapa recursos íntegros que se encuentran alojados sobre la base de datos de la plataforma CARTO.
Consultas	El objetivo de esta opción es desplegar gráficamente sobre un mapa recursos resultantes de efectuar una consulta sobre la base de datos del servidor de tripletas RDF Apache Marmotta.
Gestión de parámetros gráficos	Esta opción permite al usuario gestionar parámetros gráficos como color, ancho, transparencia e imagen de marcador. Una vez determinados son aplicados a los recursos que se grafican sobre el mapa.
Borrar Recursos	Esta opción permite remover los recursos graficados sobre el mapa.

Tabla 25: Visor operaciones disponibles

4.3.4. Parámetros de configuración

El objetivo de este archivo de configuración es centralizar la información necesaria para el correcto funcionamiento del visualizador de información geoespacial.

```

nombre_de_usuario=tomb88
clave_de_usuario=b9c60645a6d011a2e8aacc40835d0404cf0728d8
endpoint=http://190.15.132.241/marmotta/sparql
limit=1000
variable_recurso=label_respuesta
variable_wkt=wkt_respuesta
    
```

Fragmento de código 7: Visor archivo de configuración

Parámetro	Descripción
nombre de usuario	Contiene el nombre de usuario de la plataforma CARTO
clave de aplicación	Contiene la clave de aplicación de la plataforma CARTO
url Sparql Endpoint	El objetivo es conocer la dirección ip del servidor que ejecuta el servidor de tripletas RDF Apache Marmotta



Limit	El objetivo es limitar el número de recursos obtenidos después de ejecutar una consulta hacia el SPARQL Endpoint
variable recurso	Contiene el nombre del recurso tras la consulta
variable_wkt	Contiene el WKT del recurso tras la consulta
Latitud, longitud	Estos valores son necesarios para determinar el centro del mapa preferido por el desarrollador.

Tabla 26: Visor parámetros de configuración

4.3.5. Visualizar información geoespacial desde CARTO

Esta opción permite al usuario visualizar información en el contexto geoespacial ecuatoriano, esta información reside en la base de datos de la plataforma CARTO. Para conseguirlo es necesario ejecutar varias tareas previas que se detallan a continuación.

- **Determinar tipo de geometría:** Para ello es necesario ejecutar una consulta hacia la base de datos de la plataforma CARTO, esta necesita el nombre del conjunto de datos. Los posibles resultados son: punto, línea o polígono.

```
jsShowWindowLoad();
var sql = new cartodb.SQL({ user: parametro.user });
sql.execute("select ST_GeometryType(the_geom) from "+name+" WHERE cartodb_id=1 ")
.done(function(data) {
    jsRemoveWindowLoad();
    carga_parametros_grafica_desde_visualizacion(name,data.rows[0].st_geometrytype.toString());
})
.error(function(errors) {
    jsRemoveWindowLoad();
    carga_mensaje("Mensaje Informativo","No es posible establecer conexion con el servidor");
});
```

Fragmento de código 8: Tipo de geometría de recurso

- **Determinar parámetros gráficos:** Los parámetros gráficos elegidos por el usuario del visor de información geoespacial son captados en variables para su posterior uso.

```

color      =document.getElementById("color_geometria").value.toString();
transparencia=document.getElementById("trans_geometria").value.toString();
ancho      =document.getElementById("ancho_geometria").value.toString();
if(tipo=="ST_MultiPolygon" || tipo=="ST_Polygon"){
    css="#"+name+"{polygon-fill:"+color+";polygon-opacity:"+
        transparencia+";line-color:"+color+";line-width:"+ancho+";}";
}
if(tipo=="ST_MultiLineString" || tipo=="ST_LineString" ){
    css="#"+name+"{line-color:"+color+";line-opacity:"+
        transparencia+"; line-width:"+ancho+";}";
}

```

Fragmento de código 9: Captación de parámetros gráficos

- **Ejecutar consulta hacia CARTO:** Una vez que se ha determinado tanto el tipo de geometría como los parámetros gráficos elegidos por el usuario del visor de información geoespacial, el siguiente proceso es ejecutar la consulta hacia la base de datos de CARTO.

```

cartodb.createLayer(map, {user_name: parametro.user,type: 'cartodb',
sublayers: [{sql: "SELECT * FROM "+name+"",cartocss: css }]}
).addTo(map)
.on('done', function(layer) {
    cartodb.vis.Vis.addInfowindow(map, layer.getSubLayer(0), ['recurso']);
    mis_mapas[string_to_int(name)]=layer;
})
.on('error', function(err) {
    console.log("errors:" + err);
});

```

Fragmento de código 10: Ejecución consulta hacia CARTO

- **Desplegar recursos sobre el mapa:** Finalmente cuando todas las tareas antes descritas se ha ejecutado con éxito el recurso es desplegado en forma gráfica sobre el mapa.

4.3.6. Visualizar información geoespacial producto de una consulta hacia un Sparql Endpoint

Esta opción permite al usuario visualizar información en el contexto geoespacial ecuatoriano, esta información reside en la base de datos del servidor de tripletas RDF Apache Marmotta. Para conseguirlo es necesario ejecutar varias tareas previas que se detallan a continuación.

- **Determinar la consulta preferida por el usuario:** La consulta Sparql seleccionada por el usuario mediante la interfaz gráfica es captada en una variable desde un archivo que posee todas las consultas predefinidas disponibles.

```
jsShowWindowLoad();
$.ajax({type:'POST',url:'servlet_consultar',data:{numero_de_consulta:id_consulta},
success:function(result){
    jsRemoveWindowLoad();
    document.getElementById("textarea_consultas").value=result;
    dibujar_consulta();
},
error:function(error){
    jsRemoveWindowLoad();
    carga_mensaje("Mensaje Informativo","No es posible establecer conexion con el servidor");
}
});
```

Fragmento de código 11: Captación de consulta preferida por el usuario

- **Determinar parámetros gráficos:** Los parámetros gráficos elegidos por el usuario del visor de información geoespacial son captados en variables para su posterior uso.

```
css=
{
    "color":document.getElementById("color_geometria").value.toString(),
    "weight": document.getElementById("ancho_geometria").value.toString(),
    "opacity": document.getElementById("trans_geometria").value.toString()
};
```

Fragmento de código 12: Captación de parámetros gráficos

- **Ejecutar la consulta hacia Apache Marmotta:** Una vez que la consulta ha sido determinada la siguiente tarea es ejecutar la consulta SPARQL hacia el servidor de tripletas RDF específicamente en este proyecto Apache Marmotta.

```
Query query= QueryFactory.create(txt_consulta);
query.setLimit(Integer.parseInt(limit));
QueryExecution qExe;
qExe = QueryExecutionFactory.sparqlService(direccion_server,query);
ResultSet results = qExe.execSelect();
while (results.hasNext())
{
    QuerySolution binding = results.nextSolution();
    label = binding.get(var_label).toString();
    wkt = binding.get(var_wkt).toString();
    mg.add(new geometria(label,wkt));
}
```

Fragmento de código 13: Ejecución de consulta hacia Apache Marmotta

- **Desplegar recursos sobre el mapa:** Una vez que las tareas antes descritas se han ejecutado con éxito en necesario desplegar gráficamente los resultados procedentes de la consulta antes ejecutada.

```
var layergeojson=L.geoJson(JSON.parse(data_query),{
    style: css,
    onEachFeature: function(feature,layer){
        layer.bindPopup("Recurso:\n"+feature.properties.name.toString());
    },
    pointToLayer: function(feature, latlng) {
        return L.marker(latlng,{ icon:mySVGIcon });
    }
});
markers.addLayer(layergeojson);
markers.addTo(map);
```

Fragmento de código 14: Despliegue de recursos sobre el mapa

4.3.7. Gestión de parámetros gráficos

Esta opción permite al usuario gestionar parámetros gráficos como color, ancho, transparencia e imagen de marcador mediante la interfaz gráfica del visor de información geoespacial.

Parámetro	Descripción
Color	Es posible seleccionar cualquier color disponible de la paleta ofrecida por la aplicación.
Ancho	Es posible seleccionar el ancho entre los siguientes valores disponibles (1, 2, 3, 4, 5, 6).
Transparencia	Es posible seleccionar la transparencia entre los siguientes valores disponibles (0.1, 0.2,... 0.9, 1).
Imagen de marcador	Es posible elegir entre varias imágenes en formato svg, además es posible elegir el color del marcador.



Tabla 27: Visor parámetros gráficos

4.3.8. Borrar recursos desplegados sobre el mapa

Esta opción permite al usuario remover los recursos que han sido desplegados sobre el mapa.

Opción	Descripción
Remover recurso por nombre	Es posible remover recursos graficados de forma individual para ello es necesario desactivar la casilla de selección del menú recursos.
Remover todos los recursos	Es posible remover todos los recursos graficados para ello es necesario pulsar el botón correspondiente en el menú borrar recursos.
Remover todas las consultas	Es posible remover todas las consultas graficadas para ello es necesario pulsar el botón correspondiente en el menú borrar recursos.

Tabla 28: Visor opciones de borrado



Capítulo 5

5. Despliegue sobre linkeddata.ec

En el presente capítulo se describe el proceso que hizo posible la visualización, explotación y disponibilidad de descarga de información geoespacial contenida en archivos en formato RDF proveniente de la iniciativa ecuatoriana de datos enlazados. Dentro de este proceso se encuentran varias tareas como la configuración y despliegue del servidor de tripletas RDF Apache Marmotta, también fue necesaria la configuración y el despliegue del visor de información geoespacial desarrollado, finalmente para validar el funcionamiento de la herramienta de visualización desarrollada se efectuaron pruebas respecto a cada una de sus capacidades. Todas estas tareas se describen detalladamente a continuación.

5.1. Configuración y Despliegue de Apache Marmotta

El objetivo de esta sección es describir todos los procesos y tareas necesarias que permiten el uso de la plataforma Apache Marmotta. Con ello es posible gestionar la información geoespacial contenida en archivos en formato RDF, además la plataforma provee un SPARQL Endpoint que será atacado por la herramienta de visualización desarrollada.

5.1.1. Obtención y construcción de la herramienta

El proyecto de código abierto Apache Marmotta está en constante desarrollo, es posible obtenerlo en distintas versiones. En el presente proyecto hará uso de la versión 3.4.0-SNAPSHOT, esta versión provee soporte del estándar GeoSPARQL necesario para cumplir con los objetivos planteados por este proyecto.

- **Obtención de la herramienta:** Para lograrlo es necesario descargar el proyecto desde el repositorio GitHub, específicamente la rama 5-84.

Rama	MARMOTTA-584
Versión	3.4.0-SNAPSHOT
Url repositorio	https://github.com/apache/marmotta/tree/MARMOTTA-584
Documentación	https://wiki.apache.org/marmotta/MARMOTTA-584

Tabla 29: Plataforma Apache Marmotta

- **Construcción de la herramienta:** Para conseguirlo es necesario contar con Maven que nos permite construir proyectos a partir de código Java. Primero es necesario ejecutar el comando *mvn clean install* sobre la raíz del proyecto, después ejecutar el comando *mvn package* que empaqueta el código java dentro de un archivo en formato war.






	src	02/12/2016 13:43	Carpeta de archivos	
	target	02/12/2016 14:07	Carpeta de archivos	
	assembly	02/12/2016 13:43	Documento XML	3 KB
	marmotta.war	02/12/2016 14:07	Archivo WAR	39.627 KB
	pom	02/12/2016 13:44	Documento XML	26 KB

Figura 17: Construcción de Apache Marmotta

5.1.2. Despliegue de la herramienta sobre linkeddata.ec

Para desplegarlo sobre el servidor linkeddata.ec perteneciente al Departamento de Ciencias de la Computación de la Universidad de Cuenca es necesario ejecutar varias tareas que se describen a continuación.

- **Acceso al servidor de aplicaciones:** Para acceder a la interfaz gráfica del servidor de aplicaciones es necesario ingresar la url respectiva e ingresar las credenciales.

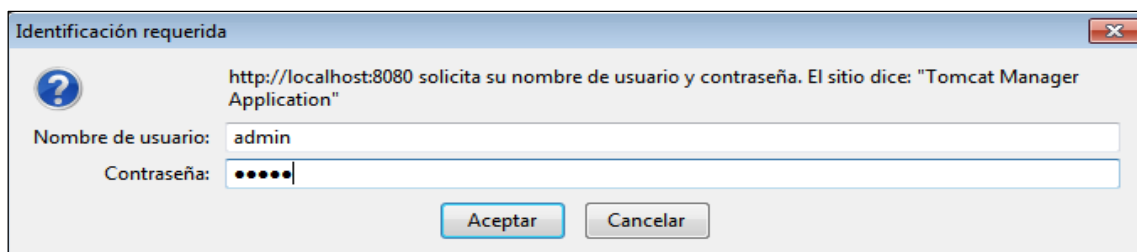


Figura 18: Acceso al servidor de Aplicaciones Web

- **Despliegue sobre el servidor de aplicaciones:** Para desplegar la aplicación es necesario acceder al administrador de aplicaciones y subirlo mediante un asistente gráfico.

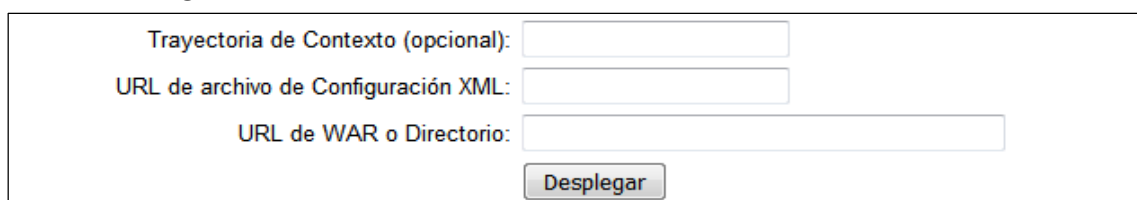


Figura 19: Despliegue de Apache Marmotta

- **Apertura de Apache Marmotta:** Para ello basta con hacer clic sobre el nombre *marmotta* que aparece sobre el administrador de aplicaciones.

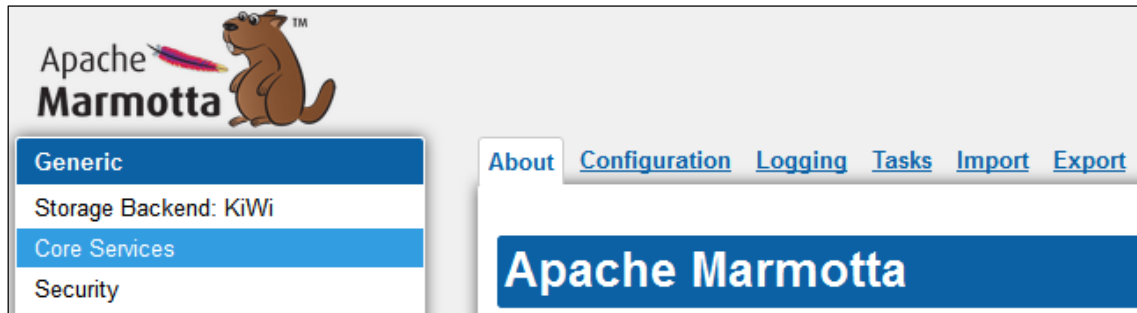


Figura 20: Interfaz Apache Marmotta

5.1.3. Configuración de la herramienta

Una vez que la herramienta Apache Marmotta se encuentra disponible sobre el servidor de aplicaciones que se ejecuta sobre el servidor *linkeddata.ec* es necesario configurarla. A continuación se describe todas las configuraciones realizadas.

- **Configuración de la base de datos:** Por defecto la herramienta está configurada con la base de datos h2, sin embargo para los propósitos del proyecto es necesario configurar una base de datos tipo PostgreSQL. Para ello es necesario ir a la subsección *database* alojada dentro de la sección *Storage Backend Kiwi* y llenar el formulario con los datos correspondientes a la instalación de PostgreSQL que se ejecuta sobre el servidor *linkeddata.ec*.

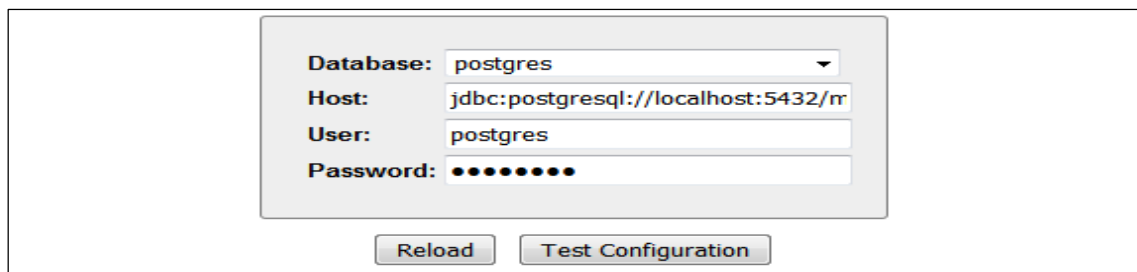


Figura 21: Configuración de PostgreSQL Apache Marmotta

- **Configuración de superusuario:** Es necesario definir dos usuarios para el correcto funcionamiento de la herramienta. El primero un usuario anónimo este podrá hacer operaciones solamente de lectura mientras que el segundo comúnmente llamado superusuario podrá hacer todo tipo de operaciones. Para ello es necesario definirlo en el archivo de configuración alojado en el directorio *home* de la plataforma Apache Marmotta.


```
kiwi.context = http://localhost:8080/marmotta/
kiwi.host = http://localhost:8080/marmotta/
kiwi.setup.host = true
user.anonymous.webid = http://localhost:8080/marmotta/user/anonymous
user.admin.webid = http://localhost:8080/marmotta/user/admin
user.admin.pwhash = :plain::passmarmotta
database.type = postgres
database.url = jdbc:postgresql://localhost:5432/marmotta?prepareThreshold=3
database.user = postgres
database.password = postgres
```

Figura 22: Configuración de Usuarios Apache Marmotta

- **Configuración de perfil de acceso:** La plataforma Apache Marmotta ofrece múltiples perfiles de seguridad, en este proyecto se estableció el perfil *simple*. A continuación se describen los perfiles que ofrece la plataforma.

Perfil	Descripción
Simple	Permite acceso de lectura a cualquier usuario, pero acceso de escritura solamente desde la red local.
Estándar	Permite acceso de lectura a cualquier usuario, pero acceso de escritura solamente a usuarios autenticados.
Restringido	Permite acceso de lectura y escritura solamente a usuarios autenticados.

Tabla 30: Perfil de acceso Apache Marmotta

5.1.4. Importación de archivos RDF hacia la herramienta

Una vez que el servidor de tripletas RDF se encuentra ejecutándose y configurado sobre el servidor de aplicaciones del servidor linkeddata.ec es necesario importar los archivos con información geoespacial en formato RDF. Para conseguirlo es necesario ejecutar varias tareas que se describen a continuación.

- **Creación del grafo geoecuator:** Apache Marmotta ofrece la posibilidad de crear varios grafos con lo que se mejora la gestión de la información. En este proyecto se creó un grafo con el nombre geoecuator que contendrá toda la información geoespacial generada por la iniciativa ecuatoriana de datos enlazados.

Label	Context
cache	http://localhost:8080/marmotta/context/cache
geoecuator	http://190.15.132.241/marmotta/context/geoecuator
W3C Linked Data Platform (LDP)	http://www.w3.org/ns/ldp#

Figura 23: Creación de grafos Apache Marmotta

- **Importar RDF hacia el grafo geoecuator:** Una vez que el grafo *geoecuator* fue creado es posible importar los archivos en formato RDF de una manera sencilla mediante la interfaz gráfica. Finalmente es posible visualizar un resumen de todos los archivos RDF que han sido importados hacia la plataforma.

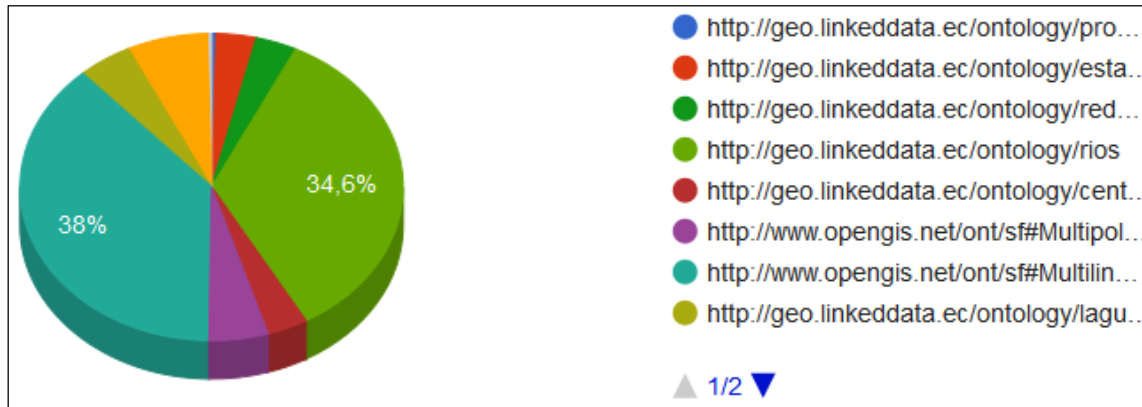


Figura 24: Importar RDF Apache Marmotta

5.2. Configuración y Despliegue del Visor de información geoespacial

El objetivo de esta sección es describir todos los procesos y tareas necesarias que permiten el uso de la herramienta de visualización y explotación de información geoespacial. Para el correcto funcionamiento del visor desarrollado es necesario que los datos geoespaciales ecuatorianos se encuentren tanto en la base de datos geoespacial de CARTO como en la base de datos de la plataforma Apache Marmotta.

5.2.1 Configuración y construcción de la herramienta

Antes de desplegar la herramienta de visualización de información geoespacial sobre el servidor linkeddata.ec es necesario identificar los parámetros correspondientes al ambiente en donde se ejecutara, luego es necesario empaquetar el código fuente en un archivo war para su posterior despliegue. A continuación se describen los procesos antes mencionados.

- **Identificación de parámetros:** Se determinaron varios parámetros relacionados a las plataformas CARTO, Apache Marmotta y linkeddata.ec. Este proceso permite el correcto funcionamiento de la herramienta de visualización de información geoespacial. A continuación se describe cada uno de estos parámetros.

Plataforma	Parámetro	Descripción
CARTO	Nombre de usuario	Necesario para realizar la gestión de la base de datos de la plataforma
CARTO	Clave de aplicación	Necesario para realizar tareas administrativas sobre la base de datos de la plataforma.
Apache Marmotta	Grafo	Necesario para realizar las consultas sobre un grado determinado.
Apache Marmotta	Url Sparql Endpoint	Necesario para conocer hacia donde se ejecutan las consultas Sparql.
Linkeddata.ec	Url	Al contener el servidor Apache Marmotta es necesario conocer la Url con fines de configuración.

Tabla 31: Identificación de parámetros Linkeddata.ec

- **Construcción de la herramienta:** Para conseguirlo es necesario contar con Maven que nos permite construir proyectos a partir de código Java. Al trabajar con el IDE Netbeans basta con seleccionar nuestro proyecto y seleccionar la opción construir con dependencias, como resultado obtendremos nuestro proyecto empaquetado en un archivo war.










	classes	24/01/2017 22:02	Carpeta de archivos	
	endorsed	09/12/2016 21:01	Carpeta de archivos	
	generated-sources	09/12/2016 21:01	Carpeta de archivos	
	maven-archiver	09/12/2016 21:01	Carpeta de archivos	
	maven-status	09/12/2016 21:01	Carpeta de archivos	
	surefire	09/12/2016 21:01	Carpeta de archivos	
	test-classes	24/01/2017 22:24	Carpeta de archivos	
	VISOR-1.0-SNAPSHOT	09/12/2016 21:01	Carpeta de archivos	
	VISOR.war	09/12/2016 21:01	Archivo WAR	25.508 KB

Figura 25: Construcción de Visor de Información geoespacial

5.2.2 Despliegue del visor sobre linkeddata.ec

Para desplegarlo sobre el servidor linkeddata.ec perteneciente al Departamento de Ciencias de la Computación de la Universidad de Cuenca es necesario ejecutar varias tareas que se describen a continuación.

- **Acceso al servidor de aplicaciones:** Para acceder a la interfaz gráfica del servidor de aplicaciones es necesario ingresar la url respectiva e ingresar las credenciales.

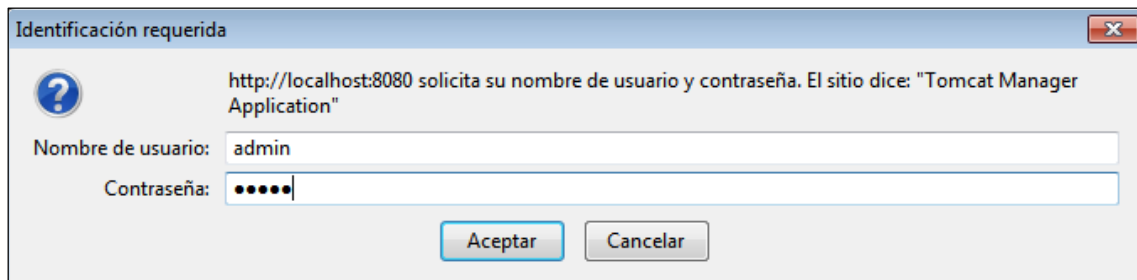


Figura 26: Acceso al servidor de Aplicaciones Web

- **Despliegue sobre el servidor de aplicaciones:** Para desplegar la aplicación es necesario acceder al administrador de aplicaciones y subirlo mediante un asistente gráfico.

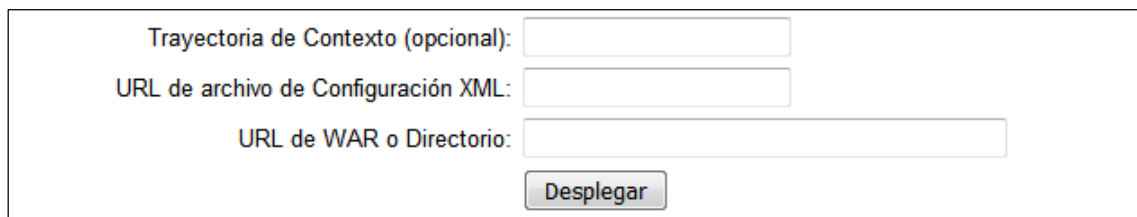


Figura 27: Despliegue de Visor de información geoespacial

Apertura del Visor de Información Geoespacial: Para ello basta con hacer clic sobre el nombre *visor* que aparece sobre el administrador de aplicaciones.

Aplicaciones			
Trayectoria	Versión	Nombre a Mostrar	Ejecutándose
/	Ninguno especificado		true
/VISOR	Ninguno especificado		true

Figura 28: Apertura del Visor de información geoespacial



5.3. Resultados

El objetivo de esta sección es describir los resultados obtenidos tras efectuar múltiples pruebas de validez y rendimiento sobre el visualizador de información geoespacial. A continuación se listan las condiciones presentes sobre las cuales se efectuaron las pruebas.

- El servidor de tripletas Apache Marmotta se encuentra ejecutándose sobre linkeddata.ec.
- El servicio de base de datos PostgreSQL ofrecido por la plataforma CARTO se encuentra habilitado y disponible.
- La aplicación desarrollada de visualización y explotación de información geoespacial se encuentra ejecutándose sobre linkeddata.ec.
- La ejecución del visor se la efectuó desde una red diferente a donde reside el servidor linkeddata.ec

5.3.1. Visualización de recursos íntegros

La visualización de recursos íntegros consiste en desplegar de forma gráfica sobre un mapa la información geoespacial contenida en un archivo en formato RDF. Previamente este archivo ha sido importado a la base de datos geoespacial de la plataforma CARTO. Finalmente el visor desarrollado es capaz de atacar esta base de datos mediante servicios web. A continuación se presentan los resultados tras efectuar múltiples pruebas relacionadas a esta opción.




Visualización de recurso “Provinciasinec”	
	
Información	
Tiempo requerido(seg)	4
Tipo de geometría	Polígono
Estado de visualización	Correcto

Tabla 32: Visualización de recurso “Provinciasinec”


Visualización de recurso “limitecontinental”	
	
Información	
Tiempo requerido(seg)	3
Tipo de geometría	Línea
Estado de visualización	Correcto

Tabla 33: Visualización de recurso “limitecontinental”



Visualización de recurso “proyectogeneracionelectrica”	
Información	
Tiempo requerido(seg)	3
Tipo de geometría	Punto
Estado de visualización	Correcto

Tabla 34: Visualización de recurso “proyectogeneracionelectrica”

5.3.2. Visualización de recursos producto de una consulta predeterminada

La visualización de recursos producto de una consulta predeterminada consiste en desplegar de forma gráfica sobre un mapa la información geoespacial resultante de efectuar una consulta determinada por el usuario hacia un SPARQL Endpoint. El visor desarrollado es capaz de atacar la base de datos de la plataforma Apache Marmotta. A continuación se presentan los resultados tras efectuar múltiples pruebas relacionadas a esta opción.



Visualización de consultas “provincias adyacentes a Pichincha”	
Información	
Tiempo requerido(seg)	4
Tipo de geometría	Polígono
Estado de visualización	Correcto

Tabla 35: Visualización de consultas “provincias adyacentes a Pichincha”

Visualización de consultas “Red vial que no cruza por el Azuay”	
Información	
Tiempo requerido(seg)	4
Tipo de geometría	Línea
Estado de visualización	Correcto

Tabla 36: Visualización de consultas “Red vial que no cruza por el Azuay”

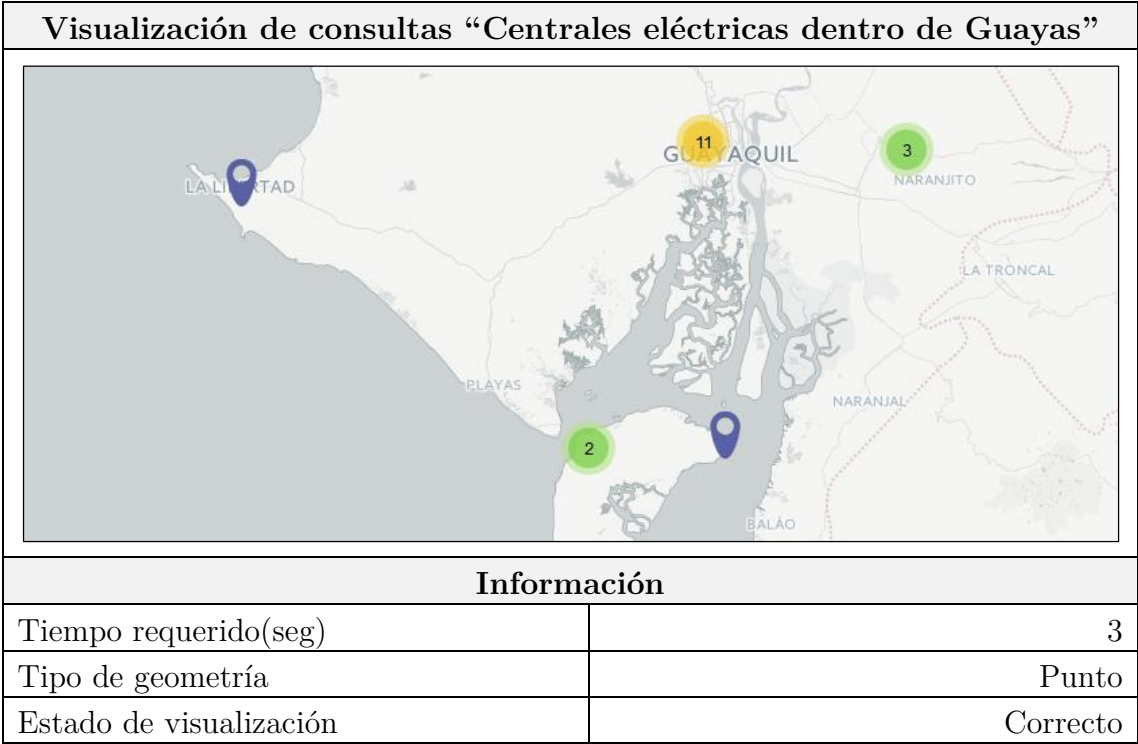


Tabla 37: Visualización de consultas “Centrales eléctricas dentro de Guayas”

5.3.3. Gestionar parámetros gráficos

La gestión de parámetros gráficos ofrece la posibilidad al usuario de visualizar recursos y consultas de forma personalizada. El usuario puede personalizar parámetros tales como color, ancho, transparencia e imagen de marcador. En las secciones referentes a pruebas de visualización de recursos y consultas se ha gestionado parámetros gráficos obteniendo los siguientes resultados.



Visualización	Parámetro especificado		Parámetro visualizado	
Visualización de recurso “provinciasinec”	Color	#547b56	Color	#547b56
	Ancho	2	Ancho	2
	Transparencia	0.2	Transparencia	0.2
	Marcador	No aplica	Marcador	No aplica
Visualización de recurso ”limitecontinental”	Color	#3e425b	Color	#3e425b
	Ancho	3	Ancho	3
	Transparencia	1	Transparencia	1
	Marcador	No aplica	Marcador	No aplica
Visualización de recurso “proyectoelctrico”	Color	#444e95	Color	#444e95
	Ancho	No aplica	Ancho	No aplica
	Transparencia	1	Transparencia	1
	Marcador	Banderín	Marcador	Banderín
Visualización de consulta “provincias adyacentes a Pichincha”	Color	#e81310	Color	#e81310
	Ancho	2	Ancho	2
	Transparencia	0.2	Transparencia	0.2
	Marcador	No aplica	Marcador	No aplica
Visualización de consulta “Red vial que no cruza por el Azuay”	Color	#2130c7	Color	#2130c7
	Ancho	3	Ancho	3
	Transparencia	1	Transparencia	1
	Marcador	No aplica	Marcador	No aplica
Visualización de consulta “Centrales eléctricas dentro de Guayas”	Color	#0a136d	Color	#0a136d
	Ancho	No aplica	Ancho	No aplica
	Transparencia	1	Transparencia	1
	Marcador	Ovalo	Marcador	Ovalo

Tabla 38: Prueba Gestión de parámetros



Capítulo 6

El objetivo de este capítulo es dar a conocer las conclusiones que se obtuvieron una vez que se ha finalizado el desarrollo de este proyecto de tesis. Estas conclusiones están alineadas a los objetivos específicos planteados en el capítulo 1 del presente documento.

6. Conclusiones

Para el desarrollo de este proyecto de tesis fueron necesarias la ejecución de varias fases, la primera de ellas consistió en la investigación de metodologías, herramientas, plataformas, formatos y tecnologías necesarias para cumplir tanto con el objetivo general como con los objetivos específicos del proyecto. Dicha investigación se relacionó específicamente con la metodología de generación y publicación de datos enlazados en el contexto geoespacial, formatos para la representación de esta información, herramientas que permitan tanto su visualización como explotación, herramientas que permitan su almacenamiento y herramientas que permitan su gestión.

La segunda fase consistió en realizar la selección de las herramientas que permitieron el desarrollo y brindaron soporte al proyecto, para ello se tomó en cuenta tanto las ventajas que presentaban unas sobre otras así como los requerimientos que se especificaron en la fase de análisis y diseño de la solución que se presentó en el capítulo 2 del presente documento.

Una vez que se determinaron las herramientas de desarrollo y soporte dio inicio la tercera fase, esta consistió en la implementación de dos aplicaciones la primera cuyo objetivo es alojar archivos en formato RDF con información geoespacial sobre la base de datos de la plataforma CARTO y la segunda cuyo objetivo es consumir y desplegar esta información de manera gráfica sobre un mapa teniendo como fuentes de información tanto la base de datos de la plataforma CARTO como la base de datos de la plataforma Apache Marmotta.

Finalmente la última fase consistió en configurar y desplegar el visor de información geoespacial desarrollado además de la plataforma Apache Marmotta sobre el servidor *Linkeddata.ec* perteneciente al departamento de Ciencias de la



Computación de la Universidad de Cuenca. Con el fin de asegurar el funcionamiento correcto del visor desarrollado se realizaron pruebas tanto de funcionalidad como de rendimiento obteniendo resultados correctos.

Todas las fases se han realizado con éxito es por ello se puede concluir que los objetivos planteados en este proyecto de tesis se han cumplido, tal como se describe a continuación.

- **Importar archivos con información geoespacial en formato RDF proveniente de la iniciativa GEO LINKED DATA ECUADOR, hacia la base de datos geoespacial de la plataforma CARTO.**

Este objetivo se cumplió mediante el desarrollo de un script que permite la importación de archivos en formato RDF que contienen información geoespacial hacia la base de datos de CARTO. Para lograrlo se utilizó como lenguaje de programación Java y varias librerías entre ellas Apache Jena que provee del lenguaje de consulta Sparql necesario para obtener la información que se encuentra contenida sobre estos archivos, GeoTools que permite la gestión de la información geoespacial en varios formatos y las APIs que provee la plataforma CARTO que permiten atacar la base de datos de la plataforma, para la seguridad de los datos fue necesario la gestión de parámetros tales como nombre usuario y clave de aplicación que son brindadas por la plataforma CARTO.

- **Visualizar información geoespacial tanto desde CARTO como desde un determinado SPARQL Endpoint (Apache Marmotta).**

Para el cumplimiento de este objetivo se desarrolló un visor de información geoespacial, este es capaz de consumir y visualizar información geoespacial de forma gráfica sobre un mapa. La información que consume proviene tanto desde la base de datos de la plataforma CARTO como de la base de datos de la plataforma Apache Marmotta. Para la construcción de este visor se utilizó como lenguaje de programación Java en el lado del servidor y JavaScript en el lado del cliente. También fue necesario el uso de varias librerías tales como Apache Jena que provee del lenguaje de consulta Sparql



necesario para atacar la base de datos de la plataforma Apache Marmotta mediante el servicio Sparql Endpoint que ofrece la plataforma, GeoTools que permite la gestión de la información geoespacial en varios formatos, las APIs que provee la plataforma CARTO que permiten atacar la base de datos de la plataforma, Carto.js que permite interactuar con los servicios de la plataforma CARTO mediante el uso de la Web y Carto.css que permite personalizar nuestras visualizaciones.

- **Explotar información geoespacial contenida en el repositorio de GEO LINKED DATA ECUADOR usando lenguaje de consulta SPARQL y su extensión GeoSPARQL.**

Para que este objetivo se cumpla fue necesario utilizar como servidor de tripletas RDF la plataforma Apache Marmotta ya que usa como lenguaje de consulta Sparql y provee soporte para el estándar GeoSparql. También fue necesario definir una serie de consultas predeterminadas que hagan uso de las funciones disponibles en el estándar GeoSparql. Mediante el visualizador desarrollado es posible ejecutar estas consultas sobre el Sparql Endpoint ofrecido por la plataforma, una vez que los resultados se encuentran disponibles en el visor, estos son desplegados gráficamente sobre un mapa. Gracias al uso del estándar GeoSparql es posible inferir conocimiento partir de información geoespacial relacionada.

El cumplimiento de cada uno de los objetivos específicos contribuye al cumplimiento del objetivo general *Visualizar y explotar información geoespacial expresada en formato RDF, proveniente del repositorio de GEO LINKED DATA ECUADOR.*

Anexos

A. Importación de archivo *cantonesAzuay.rdf* a CARTO

A continuación se describe la ejecución del script de importación de archivos en formato RDF hacia la base de datos geoespacial de CARTO. Para ello se ha seleccionado el archivo con información geoespacial referente a cantones de la provincia del Azuay.

1. **Ejecución del script de importación:** Al estar desarrollado en Java además de ser un proyecto Maven es posible ejecutarlo desde cualquier sistema operativo.

```
+-----+
SCRIPT IMPORTAR RDF GEOESPACIAL HACIA CARTO
+-----+
-----LISTA DE OPCIONES-----

A) Ver Usuario y Api Key de Acceso ha CARTO.
B) Ver Consultas SPARQL de Seleccion.
C) Probar parametros Usuario y Api Key.
D) Probar Consultas SPARQL de Seleccion.
E) Ver nombre de DataSet de CARTO.
F) Eliminar por nombre de DataSet de CARTO.
G) Importar DataSet RDF ha CARTO.
S) Salir.
Digite una opcion..
```

Figura 29: Interfaz inicial Script de importación

2. **Selección del archivo RDF:** Tras seleccionar la opción “Importar Dataset RDF ha CARTO” el script nos permite seleccionar el archivo que se requiere importar.

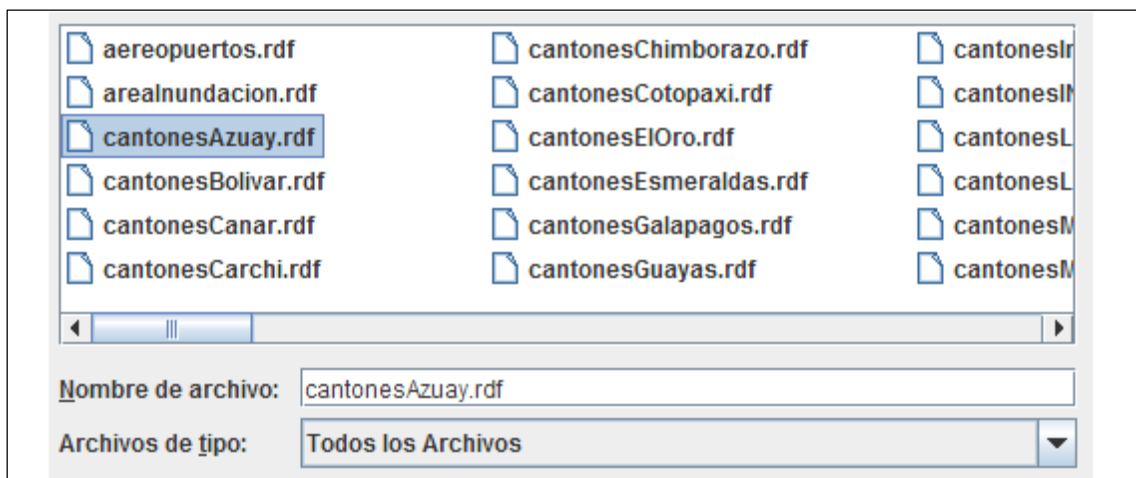


Figura 30: Selección de archivo RDF

3. **Lectura y validación de archivo RDF:** Una vez que el archivo ha sido seleccionado el script ejecuta tareas de lectura y validación de la información geoespacial contenida sobre él.

```
Leyendo Geometrias..
Recurso : LIMON INDANZA@es
Tipo : MultiPolygon
Es valida: true
Leyendo Geometrias..
Recurso : SANTIAGO@es
Tipo : MultiPolygon
Es valida: true
Leyendo Geometrias..
Recurso : NABON@es
Tipo : MultiPolygon
Es valida: true
```

Figura 31: Lectura y validación de RDF

4. **Creación e inserción de registros sobre CARTO:** Finalmente el script ejecuta tareas como la creación de la tabla, inserción de registros y operaciones para que dicha tabla pueda ser visualizada sobre el visor desarrollado.

```
Insercion numero 1 : {"rows": [], "time": 0.006, "fields": {}, "total_rows": 1}
Insercion numero 2 : {"rows": [], "time": 0.002, "fields": {}, "total_rows": 1}
Insercion numero 3 : {"rows": [], "time": 0.008, "fields": {}, "total_rows": 1}
Insercion numero 4 : {"rows": [], "time": 0.006, "fields": {}, "total_rows": 1}
Insercion numero 5 : {"rows": [], "time": 0.004, "fields": {}, "total_rows": 1}
Insercion numero 6 : {"rows": [], "time": 0.014, "fields": {}, "total_rows": 1}
Insercion numero 7 : {"rows": [], "time": 0.014, "fields": {}, "total_rows": 1}
Insercion numero 8 : {"rows": [], "time": 0.006, "fields": {}, "total_rows": 1}
Insercion numero 9 : {"rows": [], "time": 0.007, "fields": {}, "total_rows": 1}
Insercion numero 10 : {"rows": [], "time": 0.008, "fields": {}, "total_rows": 1}
Insercion numero 11 : {"rows": [], "time": 0.006, "fields": {}, "total_rows": 1}
```

Figura 32: Creación e inserción de registros sobre CARTO

5. **Información alojada sobre CARTO:** Cuando todas las tareas antes descritas se han ejecutado con éxito entonces es posible consumir la información importada desde el visor de información geoespacial desarrollado.

cartodb_id number	the_geom geometry	recurso string	
1	Polygon	LIMON INDANZA@es	
2	Polygon	SANTIAGO@es	
3	Polygon	NABON@es	
4	Polygon	PAUTE@es	

Figura 33: Información geoespacial sobre CARTO

B. Eliminación de archivo *cantonesAzuay.rdf* de CARTO

A continuación se describe la ejecución del script de importación de archivos en formato RDF hacia la base de datos geoespacial de CARTO. Se describe la operación que permite la eliminación de un conjunto de datos para ello es necesario especificar el nombre del mismo.

1. **Ejecución del script de importación:** Al estar desarrollado en Java además de ser un proyecto Maven es posible ejecutarlo desde cualquier sistema operativo.

```
+-----+
SCRIPT IMPORTAR RDF GEOESPACIAL HACIA CARTO
+-----+
-----LISTA DE OPCIONES-----

A) Ver Usuario y Api Key de Acceso ha CARTO.
B) Ver Consultas SPARQL de Seleccion.
C) Probar parametros Usuario y Api Key.
D) Probar Consultas SPARQL de Seleccion.
E) Ver nombre de DataSet de CARTO.
F) Eliminar por nombre de DataSet de CARTO.
G) Importar DataSet RDF ha CARTO.
S) Salir.
Digite una opcion..
```

Figura 34: Interfaz inicial Script de importación

2. **Petición de nombre de Dataset:** Tras seleccionar la opción “Eliminar por nombre de DataSet de CARTO” el script requiere el nombre del DataSet y después ejecuta tareas para eliminarlo, una vez que se ha eliminado ya no es posible visualizarlo desde el visor desarrollado.

```
+-----+
+-----ELIMINAR DATASET CARTO-----+
+-----+
Ingrese el nombre de dataset a eliminar :
cantonAzuay
Dataset eliminado correctamente
```

Figura 35: Eliminación de conjunto de datos

C. Enumeración de conjuntos de datos sobre CARTO.

A continuación se describe la ejecución del script de importación de archivos en formato RDF hacia la base de datos geoespacial de CARTO. Se describe la operación que permite la enumeración de los conjuntos de datos alojados sobre la base de datos de CARTO.

1. **Ejecución del script de importación:** Al estar desarrollado en Java además de ser un proyecto Maven es posible ejecutarlo desde cualquier sistema operativo.

```
+-----+
SCRIPT IMPORTAR RDF GEOESPACIAL HACIA CARTO
+-----+
-----LISTA DE OPCIONES-----

A) Ver Usuario y Api Key de Acceso ha CARTO.
B) Ver Consultas SPARQL de Seleccion.
C) Probar parametros Usuario y Api Key.
D) Probar Consultas SPARQL de Seleccion.
E) Ver nombre de DataSet de CARTO.
F) Eliminar por nombre de DataSet de CARTO.
G) Importar DataSet RDF ha CARTO.
S) Salir.
Digite una opcion..
```

Figura 36: Interfaz inicial Script de importación

2. **Enumeración de conjuntos de datos:** Tras seleccionar la opción “Ver nombre de DataSet de CARTO” el script lista todos los conjuntos de datos que se encuentran alojados sobre la base de datos de CARTO.

```
+-----+
+-----VER DATASET CARTO-----+
+-----+

Tabla : cantonescanar
Tabla : parquenacionalimae
Tabla : cantonescotopaxi
Tabla : cuencasmagap
Tabla : carreteras
Pulse INTRO para continuar..
```

Figura 37: Enumeración de conjunto de datos alojados sobre CARTO



Acrónimos

AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
BSD	Berkeley Software Distribution
CSS	Cascading Stylesheets
DCC	Departamento de Ciencias de la Computación
GEOJSON	GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON)
GEOSPARQL	GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data
GIS	Geographic information system
GML	Geography Markup Language
HP	Hewlett-Packard
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IGM	Instituto Geográfico Militar
JS	Java Script
JSON	JavaScript Object Notation
OGC	Open Geospatial Consortium
OWL	Ontology Web Language
ORDBMS	Object-relational database management system



PNG	Portable Network Graphics
PROMAS	Programa para el manejo del Agua y el Suelo
REST	Representational state transfer
RDBMS	Relational database management system
RDF	Resource Description Framework
RDFa	Resource Description Framework in Attributes
RDFs	Resource Description Framework Schema
SENAGUA	Secretaria Nacional del Agua
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SVG	Scalable Vector Graphics
UML	Unified Modeling Language
URI	Uniform Resource Identifier
UWE	UML-based web engineering
WKT	Well Known Text
W3C	World Wide Web Consortium
XML	Extensible Markup Language



Bibliografía

- Auer, S., Lehmann, J., & Hellmann, S. (2009). Linkedgeodata: Adding a spatial dimension to the web of data. In *International Semantic Web Conference* (pp. 731–746). Springer.
- Battle, R., & Kolas, D. (2012). Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web*, 3(4), 355–370.
- Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205–227.
- Castells, P. (2003). La web semántica. *Sistemas Interactivos Y Colaborativos En La Web*, 195–212.
- Gómez-Pérez, A. (2012). Linked data applications: examples.
- González Balea, R. (2012). Análisis de plataformas para la publicación de información geográfica en la nube. Universitat Oberta de Catalunya.
- Halevy, A., & Shapley, R. (2009). Google fusion tables. *Research Blog*, June, 9.
- Hartig, O., Bizer, C., & Freytag, J.-C. (2009). Executing SPARQL queries over the web of linked data. In *International Semantic Web Conference* (pp. 293–309). Springer.
- Jena, A. (2007). semantic web framework for Java.
- Jena, J., & Fuseki, P. (2004). Semantic web frameworks. *Oaks, R., University of Sterling. Disponible En Http://www. Cs. Stir. Ac. uk/courses/31Z7/posters/2014/rao. Pdf.*
- Lassila, O., & Swick, R. R. (1999). Resource description framework (RDF) model and syntax specification.
- Leon, A. de, Wisniewski, F., Villazón-Terrazas, B., & Corcho, O. (2012). Map4rdf-faceted browser for geospatial datasets.
- Llaves, A., Corcho, O., & Fernandez-Carrera, A. (2014). Map4rdf-ios: a tool for exploring linked geospatial data. In *Proceedings of Workshop on Linked Geospatial Data*.
- Macdonald, S. (2008). Data visualisation tools: Part 2-spatial data in a web 2.0 environment and beyond.



- Mader, C., Martin, M., & Stadler, C. (2014). Facilitating the Exploration and Visualization of Linked Data (pp. 90–107). https://doi.org/10.1007/978-3-319-09846-3_5
- Montero, Y. H., Viedma, E. H., Torres, J. C. H., & Redondo, E. P. (2003). Análisis de la web semántica: estado actual y requisitos futuros. *El Profesional de La Información*, 12(5), 368–376.
- Musciano, C., & Kennedy, B. (1996). *HTML, the definitive Guide*. O'Reilly & Associates.
- Nikolaou, C., Dogani, K., Kyzirakos, K., & Koubarakis, M. (2013). Sextant: browsing and mapping the ocean of linked geospatial data. In *Extended Semantic Web Conference* (pp. 209–213). Springer.
- Novillo, R. L. L. (2016). Aporte metodológico para el monitoreo de recursos hídricos utilizando tecnología semánticas: caso de estudio: conflación semántica de información geográfica en el dominio hídrico ecuatoriano. Universidad Pontificia de Salamanca.
- Provider, V. J. (2009). OpenLink Virtuoso Universal Server: Documentation. *OpenLink Software*.
- Tello, A. L. (2001). Ontologías en la Web semántica. *Jornadas de Ingeniería Web*, 1.
- Turton, I. (2008). Geo tools. In *Open source approaches in spatial data handling* (pp. 153–169). Springer.
- Vilches-Blázquez, L. M. ;, Sevilla, C. ;, Villalón, M. ;, Rodríguez, A. F. ;, & Gómez-Pérez, A. (n.d.). Combinando Linked Data con servicios geoespaciales.
- Villazón-Terrazas, B., & Corcho, O. (2011). Methodological guidelines for publishing linked data. *Una Profesión, Un Futuro: Actas de Las XII Jornadas Españolas de Documentación: Málaga*, 25(26), 20.
- Volz, J., Bizer, C., Gaedke, M., & Kobilarov, G. (2009). Silk-A Link Discovery Framework for the Web of Data. *LDOW*, 538.